Universidad de El Salvador
Facultad de Ciencias Naturales y Matemática
Escuela de Matemática

Trabajo de grado titulado

Agrupamiento para la Identificación de Modelos Difusos

Presentado por

Br. Wilfredo Gómez Melara

Br. Ofelia Janeth Valladares Martínez

Para optar al grado de

Licenciado en Matemática

Ciudad universitaria, 3 de mayo de 2023

Universidad de El Salvador
Facultad de Ciencias Naturales y Matemática
Escuela de Matemática

Agrupamiento para la Identificación de Modelos Difusos

Presentado por:

Br.Wilfredo Gómez Melara

Br. Ofelia Janeth Valladares Martínez

Para optar al grado de

Licenciado en Matemática

Bajo la dirección del

MSc. Carlos Ernesto Gámez Rodríguez.

Ciudad universitaria, 3 de mayo de 2023.

Universidad de El Salvador

Autoridades

MSc. Roger Armando Arias

Rector

PhD. Raúl Ernesto Azcúnaga López

Vicerrector Académico

tari Ing. Juan Rosa Quintanilla

Vicerrector Administrativo

MSc. Francisco Antonio Alarcón Sandoval

Secretario General

Lic. Rafael Humberto Peña Marín

Fiscal General

Lic. Luis Antonio Mejía Lipe

Defensor de los Derechos Universitarios

Facultad de Ciencias Naturales y Matemática

Autoridades

Lic. Mauricio Hernán Lovo Córdoba

Decano

MSc. Zoila Virginia Guerrero Mendoza

Vicedecana

Lic. Jaime Humberto Salinas Espinoza

Secretario de la Facultad

Escuela de Matemática

Dr. Dimas Noé Tejada Tejada

Director

MSc. Carlos Ernesto Gámez Rodríguez

Secretario

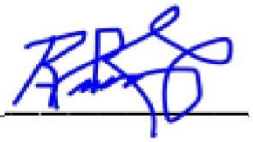
Tribunal Calificador

MSc. Carlos Ernesto Gámez Rodríguez _____

Asesor de Tesis

MSc. René Armando Peña Aguilar _____

Jurado



MSc.Porfirio Armando Rodríguez Rodríguez _____

Jurado

Dedicatoria

Dedicamos el resultado de este trabajo a nuestro optimismo y perseverancia, porque a pesar de haber tenido muchas dificultades, hemos logrado salir adelante sin perder nunca la cabeza ni morir en el intento.

Agradecimientos

Agradecemos a Dios por concedernos la fuerza y todos los elementos necesarios para resolver nuestras dificultades y aprender de ellas.

Agradecemos al Msc. Carlos Ernesto Gámez por toda su asesoría y apoyo en el desarrollo de este trabajo.

Wilfredo Gómez

Agradezco a mis padres Luis Gómez y Zoila Melara por su apoyo y buenos consejos a lo largo de toda mi carrera.

Janeth Valladares

Agradezco a mi madre Félix María Martínez y a mis hermanos Rafael y Alma Valladares por su apoyo y comprensión.

Agradezco a mi compañero Wilfredo Gómez por tenerme paciencia a lo largo del desarrollo de este trabajo y brindarme su ayuda para poder terminarlo.

Índice general

Introducción

Muchos procesos reales contienen complejas relaciones no lineales y variantes en el tiempo, difíciles de modelar. Esto ha llevado a la investigación y desarrollo de métodos y herramientas sofisticadas de construcción de sistemas inteligentes. Tales sistemas deben ser rápidos, adaptables, no lineales y capaces de captar la dinámica del modelo.

En los últimos años se ha prestado especial atención a las técnicas de manejo de datos para la generación de modelos flexibles entre las que se encuentran los sistemas difusos. Teóricamente se ha demostrado que bajo ciertas condiciones, un sistema difuso se comporta como un aproximador universal. Dentro de los sistemas difusos, el modelo Takagi Sugeno (TS) se ha convertido en una herramienta práctica y potente para el modelado de sistemas complejos, debido a que es capaz de describir sistemas altamente no lineales utilizando un pequeño número de reglas.

Gran parte de los métodos para la obtención de modelos difusos auto-organizados utilizan métodos de agrupamiento para seccionar el espacio de datos de entrada-salida, combinados con algoritmos genéticos, mínimos cuadrados u optimización del tipo gradiente descendente. Sin embargo, no son muchos los que proporcionan un verdadero proceso de adaptación. La construcción de un modelo borroso requiere de la selección de un gran número de parámetros: número, forma y

distribución de las funciones de pertenencia, construcción de la base de reglas, selección de operadores lógicos, selección del método de inferencia, entre otros, lo cual requiere de criterios sistemáticos para una acertada selección. Los algoritmos de agrupamiento difuso representan la técnica más adecuada para la obtención de modelos difusos, siendo los métodos de Fuzzy C-Means y de Gustafson Kessel los más empleados.

El presente trabajo se ha desarrollado de la siguiente forma: los primeros dos capítulos conforman el marco teórico, el tercer capítulo consiste en una serie de ejemplos en los que se muestra la aplicación de algunos algoritmos de agrupamiento difuso.

En el capítulo I, se da una introducción profunda sobre el agrupamiento, enfatizando los métodos y algoritmos que se usan en el resto del trabajo. En aras de la exhaustividad, también se presenta una breve descripción general de otros métodos. Este capítulo brinda una descripción detallada sobre el agrupamiento borroso con ejemplo para ilustrar la diferencia entre ellos. Ademas en conexión directa con el agrupamiento: se trata la visualización de los resultados del agrupamiento. Los métodos presentados permiten al lector ver los grupos n-dimensionales, por lo tanto, validar los resultados.

El capítulo II, trata sobre la identificación de modelos difusos y presenta métodos para resolverlos. La familiaridad con la regresión y el modelado es útil pero no necesaria porque habrá una descripción general de los conceptos básicos del modelamiento difuso en la introducción.

El capítulo III, se muestran algunos ejemplos prácticos, en donde se puede observar que el modelamiento difuso se puede emplear en diferentes areas de la vida, de tal manera de que no dista de la realidad.

Capítulo 1

Marco teórico

En esta sección se agregará la teoria pertinente al tema de agrupamiento difuso y los algoritmos de identificación de patrones más conocidos.

1.1. Análisis de clúster difuso clásico

1.1.1. Tipos de datos

Los datos pueden ser relativos o absolutos. Datos relativos significa que sus valores no lo son, pero se conocen sus distancias en pares. Estas distancias se pueden organizar como una matriz llamada matriz de proximidad. También se puede ver como un gráfico ponderado. En ste trabajo se considera principalmente "datos absolutos", por lo que queremos dar algunas expresiones más precisas sobre esto.

Los tipos de datos absolutos se pueden organizar en cuatro categorías. Sean x y x' dos valores del mismo atributo.

- Tipo nominal. En este tipo de datos, lo único que se puede decir acerca de dos datos es si son iguales o no: $x = x'$ ó $x \neq x'$.
- Tipo ordinal. Los valores se pueden organizar en una secuencia. Si $x \neq x'$, entonces también es verificable que $x < x'$ ó bien $x > x'$

- Escala de intervalo. Si la diferencia entre dos elementos de datos puede expresarse como un número además de los términos mencionados anteriormente.

- Escala de relación. Este tipo de datos es una escala de intervalo pero también existe un valor cero. Si $c = x/x'$, entonces se puede decir que x es c veces más grande que x'

En este trabajo, se considera la agrupación de datos de escala de relación. Los datos son típicamente observaciones de algunos fenómenos. En estos casos, no solo se miden una, sino n variables, por lo tanto, cada observación consta de n variables medidas, agrupadas en un vector de columna n-dimensional $x_k = [x_{1,k}, x_{2,k}, x_{3,k}, ..., x_{n,k}]^T$, $x_k \in \mathbb{R}^n$. Estas variables no suelen ser independientes unas de otras, por lo tanto se necesita un análisis de datos multivariado que sea capaz de manejar estas observaciones. Un conjunto de N observaciones se denota por $X = \{x_k | k = 1, 2, ..., N\}$, y es representado con una matriz de dimensión $N \times n$:

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,n} \end{bmatrix} \tag{1.1}$$

En la terminología de reconocimiento de patrones, las filas de X se denominan patrones u objetos, las columnas se denominan características o atributos y X se llama matriz de patrones. En éste trabajo, a menudo se hace referencia a X simplemente como la matriz de datos. El significado de las filas y columnas de X con respecto a la realidad depende del contexto. En el diagnóstico médico, por ejemplo, las filas de X pueden representar a los pacientes, y las columnas son síntomas

o mediciones de laboratorio para los pacientes. Cuando se aplica la agrupación en clúster al modelado y la identificación de sistemas dinámicos, las filas de X contienen muestras de señales de tiempo, y las columnas son, por ejemplo, variables físicas observadas en el sistema (posición, velocidad, temperatura, etc.). En la identificación del sistema, el propósito de la agrupación en clústers es encontrar relaciones entre las variables del sistema independientes, llamadas regresores, y los valores futuros de las variables dependientes, llamados regresiones. Sin embargo, uno debe darse cuenta de que las relaciones reveladas por la agrupación son solo asociaciones de acausal entre los vectores de datos, y como tales aún no constituyen un modelo de predicción del sistema dado.

Los datos se pueden dar en forma de la llamada matriz de disimilitud:

$$\begin{bmatrix} 0 & d(\,1,\,2) & d(\,1,\,3) & \cdots & d(\,1,\,N) \\ & 0 & d(\,2,\,3) & \cdots & d(\,2,\,N) \\ & & 0 & \ddots & \vdots \\ & & & & 0 \end{bmatrix} \tag{1.2}$$

donde $d(i,j)$ significa la medida de disimilitud (distancia) entre el objeto x_i y x_j. Debido a que $d(i,i) = 0, \forall i$, se pueden encontrar ceros en la diagonal principal, y esa matriz es simétrica porque $d(i,j) = d(j,i)$. Hay algoritmos de agrupación en clústers que utilizan esa forma de datos (por ejemplo, métodos jerárquicos). Si los datos se proporcionan en forma de (1.1), el primer paso que se debe hacer es transformar los datos en una matriz de disimilitud.

1.1.2. Medidas de Similitud

Dado que la similitud es fundamental para la definición de un grupo, una medida de la similitud entre dos patrones extraídos del mismo espacio de características es esencial para la mayoría de los procedimientos de agrupamiento.

Debido a la variedad de tipos de características y escalas, la medida de la distancia (o medidas) debe elegirse con cuidado. Es más común calcular la diferencia entre dos patrones utilizando una medida de distancia definida en el espacio de la característica. Nos centraremos en las medidas de distancia conocidas utilizadas para patrones cuyas características son todas continuas.

La métrica más popular para características continuas es la **distancia euclidiana**

$$d_2(\ x_i,\ x_j) = \left(\sum_{k=1}^{d}(x_{i,k} - x_{j,k})^2\right)^{\frac{1}{2}} = \|x_i - x_j\|_2 \tag{1.3}$$

La cual es un caso especial $(p = 2)$ de la **métrica de Minkowski**

$$d_p(\ x_i,\ x_j) = \left(\sum_{k=1}^{d}|x_{i,k} - x_{j,k}|^p\right)^{\frac{1}{p}} = \|x_i - x_j\|_p \tag{1.4}$$

La distancia euclidiana tiene un atractivo intuitivo, ya que se usa comúnmente para evaluar la proximidad de los objetos en un espacio de dos o tres dimensiones. Funciona bien cuando un conjunto de datos tiene grupos compactos o aislados. El inconveniente del uso directo de las métricas de Minkowski es la tendencia de la característica de mayor escala a dominar a las demás. Las soluciones a este problema incluyen la normalización de las características continuas (a un rango o varianza común) u otros esquemas de ponderación. La correlación lineal entre las características también puede distorsionar las medidas de distancia; esta distorsión se puede aliviar aplicando una transformación de blanqueamiento a los datos o utilizando el cuadrado de la **Distancia de Mahalanobis**.

$$d_M(x_i, x_j) = (x_i - x_j)F^{-1}(x_i - x_j)^T \tag{1.5}$$

Donde se asume que los patrones x_i y x_j son vectores de fila, y F es la matriz de covarianza de muestra de los patrones o la matriz de covarianza conocida del

proceso de generación de patrón; $d_M(\cdot,\cdot)$ asigna diferentes pesos a diferentes características en función de sus varianzas y correlaciones lineales por pares. Aquí, se asume implícitamente que las densidades condicionales de clase son unimodales y se caracterizan por una propagación multidimensional, es decir, que las densidades son gaussianas multivariadas.

Algunos algoritmos de agrupación trabajan en una matriz de valores de proximidad en lugar de en el conjunto de patrones original. Es útil en tales situaciones para calcular previamente todos los valores de distancia por pares $\frac{N(N-1)}{2}$ para los patrones N y almacenarlos en una matriz (simétrica). El cálculo de distancias entre patrones con algunas o todas las características no continuas es problemático, ya que los diferentes tipos de características no son comparables y (como un ejemplo extremo) la noción de proximidad tiene un valor binario efectivo para las características de escala nominal. No obstante, los profesionales (especialmente aquellos en aprendizaje automático, donde los patrones de tipo mixto son comunes) han desarrollado medidas de proximidad para patrones de tipo heterogéneos. Un ejemplo reciente propone una combinación de una métrica modificada de Minkowski para características continuas y una distancia basada en conteos (población) para atributos nominales.

1.1.3. Técnicas de agrupamiento

Durante toda su historia, el hombre ha obtenido conocimiento a partir de la clasificación de objetos. La sociedad actual se caracteriza por la cantidad de información a su alcance. El uso de ordenadores y el incremento de la capacidad de almacenaje permiten que cada vez se guarden más datos. Toda esta información es procesada y analizada para obtener de ella alguna información relevante, que pueda ser manejada para uso propio o para futuros análisis.
Para poder comprender un objeto, las personas tienden a caracterizarlo y a compa-

rarlo con otros objetos existentes, basándose en su similitud o su disimilitud. Para poder caracterizarlo y compararlo es necesario disponer de la máxima cantidad de información.

Uno de los métodos de clasificación de datos es el agrupamiento. Este consiste en ordenar observaciones o vectores de características en grupos (clusters), sin tener ningún tipo de información sobre la salida, esto es, sin disponer de datos etiquetados. Al no poseer ningún conocimiento acerca de los patrones de salida, nuestro sistema de clasificación tiene que descubrir la estructura interna de similitud de los datos de forma eficiente. El agrupamiento es útil en muchas técnicas exploratorias de análisis de patrones, minería de datos, toma de decisiones,... es decir, es útil en muchas técnicas de aprendizaje automático. Sin embargo, en muchos de estos problemas existe poca información a priori (como sucede por el contrario en los modelos estadísticos) y a la hora de tomar las decisiones se deben hacer el mínimo número posible de asunciones sobre los datos.

Podemos definir como agrupamiento al proceso de agrupar objetos,de tal modo que los objetos de un mismo grupo son más similares unos a otros que con objetos de otros grupos.

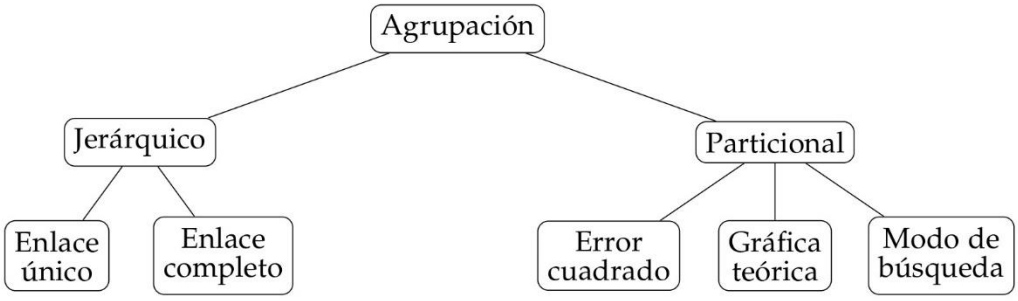
Definición 1 *Dado un conjunto de datos de entrada* $X = [x_1, x_2, \cdots, x_N]$*, tal que* $x = [x_{i1}, x_{i2}, \cdots, x_{id}]^T$*, siendo cada componente* $x_{ij} \in \mathbb{R}^d$ *una característica distinta, se define una m-agrupación de X a una partición del conjunto de datos X en m conjuntos o clusters* $C = C_1, C_2, \ldots, C_m$*, de tal manera que se cumplan estas tres condiciones:*

- $C_i \neq \emptyset, i = 1, 2, \ldots, m$
- $\cup_{i=1}^{m} C_i = X$
- $C_i \cap C_j = \emptyset, \forall i, j : i \neq j, i, j = 1, \ldots, m$

El conjunto de vectores contenidos en C_i *son más similares entre ellos y menos similares entre los contenidos en* C_j*.*

Según esta definición dada, a cada objeto o punto se le asigna una única clase, grupo o cluster, es decir, la agrupación llevada a cabo es de tipo dura.

Los diferentes enfoques para los datos de agrupamiento se pueden describir con la ayuda de la jerarquía siguiente:



Algoritmos de agrupamiento jerárquico

Los algoritmos de agrupamiento jerárquicos organizan los datos en estructuras jerárquicas de acuerdo a la matriz de proximidades. El proceso de agrupamiento comienza suponiendo que existe una secuencia de particiones de n objetos en K grupos. La primera de estas particiones es una partición en n grupos, cada uno conteniendo exactamente un objeto. La siguiente es una partición en $n-1$ grupos, la siguiente en $n-2$, y así sucesivamente hasta la partición n-ésima en la cual, todos los objetos forman un grupo. Se dice que nos encontramos en el nivel c de una secuencia cuando $K = n - c + 1$. Así, el primer nivel corresponde a n grupos y el nivel n corresponde a un grupo. Dados dos objetos x_1 y x_2, en algún nivel estarán agrupados juntos en el mismo grupo. Si esta secuencia cumple la propiedad de que siempre que dos objetos se encuentran en un mismo grupo en el nivel c y que siguen estando juntos en el resto de niveles, entonces esta secuencia es un agrupamiento jerárquico.

Un algoritmo jerárquico produce un dendrograma que representa la agrupación anidada de patrones y niveles de similitud en los que cambian las agru-

paciones. El dendrograma se puede romper en diferentes niveles para obtener diferentes agrupamientos de los datos.

Algoritmos de partición

Un algoritmo de agrupamiento particional es aquel que obtiene como resultado una única partición de los datos iniciales, en lugar de una estructura de agrupamiento con varios niveles de particiones. Un algoritmo particional asigna a un conjunto de objetos K grupos sin estructura jerárquica, siendo K un número real menor que el número total de objetos. Este tipo de algoritmos son muy eficientes en aquellas aplicaciones con conjuntos de datos de gran dimensionalidad, pero presentan el problema de que es necesario escoger el número de grupos deseados.

Los algoritmos particionales por lo general producen grupos optimizando una función objetivo definida, bien localmente (definida sobre un subconjunto de datos) o bien globalmente (definida sobre el conjunto completo de datos). Una forma de encontrar la partición óptima es através de una búsqueda combinatoria del conjunto de valores de etiquetas posibles, pero esta solución es computacionalmente inviable. Por lo general, en vez de esta búsqueda prohibitiva, se opta por ejecutar varias veces el algoritmo con distintos puntos de entrada y la mejor configuración obtenida de entre todas las ejecuciones es usada como salida del algoritmo.

Uno de los factores más importantes en los algoritmos particionales es la función objetivo. En los algoritmos particionales la función objetivo más utilizada es el error cuadrático.La expresión (1.6) define el criterio de error cuadrático dado un conjunto de entrada $x_j \in \mathbb{R}^d, j = 1, 2, \ldots, N$ que se quiere agrupar en un conjunto de K grupos, $C = C_1, C_2, \ldots, C_k$ en esta ecuación, Γ es una matriz de particiones y M es una matriz de medias, centroides o prototipos de grupos. Los grupos resultantes de esta función de error cuadrática son frecuentemente denominados particiones de varianza mínima.

$$J(\Gamma, M) = \sum_{i=1}^{K} \sum_{j=1}^{N} \gamma_{ij} \|x_j - m_i\|^2 \tag{1.6}$$

Donde $\Gamma = [\gamma_{ij}]$ es la matriz de elementos, siendo

$$\gamma_{ij} = \begin{cases} 1 & \text{si } x_j \in \textit{cluster } i \\ 0 & \text{otro} \end{cases}$$

Con

$$\sum_{i=1}^{K} \gamma_{ij} = 1, \forall j;$$

y donde $M = [m_1, \ldots, m_K]$ es la matriz de medias, siendo $m_i = \frac{1}{N_i} \sum_{j=1}^{N} \gamma_{ij} x_j$ una muestra de la media del grupo i dados N_i objetos en ese grupo.

Existen numerosos algoritmos particionales basándose en distintos aspectos, como por ejemplo: en las funciones objetivo, en los atributos de entrada, en el tipo de salida, etc. Uno de los algoritmos más utilizados y más sencillos son: K-Means Clustering. Se basa en el mismo principio fundamental. Pero, es un algoritmo duro, es decir, a cada objeto se le asigna un único grupo. A continuación se explica con más detenimiento.

Algoritmo k-means

K-means es el algoritmo comúnmente utilizado que emplea un criterio de error cuadrado. Comienza con una partición inicial aleatoria y continúa reasignando los patrones a los clústers en función de la similitud entre el patrón y los centros del clúster hasta que se cumple un criterio de convergencia (por ejemplo, no hay ninguna reasignación de ningún patrón de un clúster a otro, o el cuadrado) el error deja de disminuir significativamente después de un cierto número de iteraciones). El algoritmo k-means es popular porque es fácil de implementar y su complejidad de tiempo es $O(N)$, donde N es el número de patrones. Un problema importante con este algoritmo es que es sensible a la selección de la partición inicial y puede

converger a un mínimo local del valor de la función de criterio si la partición inicial no se elige correctamente.El procedimiento completo se sigue de la siguiente manera:

Algoritmo k-means

1.**Inicialización:** Una vez escogido el número de grupos, k, se establecen k centroides en el espacio de los datos, por ejemplo, escogiéndolos aleatoriamente.

2.**Asignación objetos a los centroides:** Cada objeto de los datos es asignado a su centroide más cercano.

3.**Actualización de centroides:**Se actualiza la posición del centroide de cada grupo tomando como nuevo centroide la posición del promedio de los objetos pertenecientes a dicho grupo.

4.Repetir 2 y 3 hasta que no haya cambios en cada cluster

1.1.4. Agrupamiento difuso

Dado que los grupos pueden verse formalmente como subconjuntos del conjunto de datos, una posible clasificación de los métodos de agrupamiento puede ser si los subconjuntos son difusos o deterministas(duro). Los métodos de agrupamiento duro se basan en la teoría de conjuntos clásica y requieren que un objeto pertenezca o no al agrupamiento. Pero también podría darse el caso de que un vector pudiera pertenecer a varias clases con un cierto grado de pertenencia, es el denominado $fuzzy$ clustering. Un agrupamiento tipo $fuzzy$ del conjunto de datos X en c grupos, se caracteriza por c funciones μ_i, denominadas funciones de pertenencia, donde:

$$\mu_i : X \to [0,1], i = 1, \ldots, c$$

y

$$\sum_{i=1}^{c} \mu_i(x_k) = 1, k = 1, \ldots, N$$

Y teniendo en cuenta que

$$0 \leq \sum_{k=1}^{N} \mu_i(x_k) \leq N, i = 1, \ldots, c$$

El valor de estas funciones de pertenencia es una caracterización matemática de un conjunto, en nuestro caso, el grupo. Aquellos valores más cercanos a la unidad representan un mayor grado de pertenencia a un grupo, y aquellos cercanos al cero representan un menor grado de pertenencia.

Los métodos de agrupamiento difuso permiten que los objetos pertenezcan a varios clústers simultáneamente, con diferentes grados de pertenencia. El conjunto de datos X se divide así en c subconjuntos difusos. **En nuestras situaciones reales, el agrupamiento difuso es más natural que el agrupamiento duro, ya que los objetos en los límites entre varias clases no están obligados a pertenecer completamente a una de las clases, sino que se les asigna grados de pertenencia entre** 0 **y** 1 **que indica su pertenencia parcial.**

La naturaleza discreta de la partición dura también causa la intratabilidad analítica y algorítmica basados en funciones analíticas, ya que estas funciones no son diferenciables. Acontinuación definiremos mas precisamente el concepto de particiones difusas.

Definición 2 *Sea* $X = [x_1, x_2, \cdots, x_N]$ *un conjunto finito y sea* $2 \leq c < N$ *un número entero. Definimos el conjunto:*

$$M_{fco} = \left\{ U \in \mathbb{R}^{c \times N} \mid \mu_{i,k} \in [0,1], \forall i,k; \sum_{i=1}^{c} \mu_{i,k} = 1, \forall k; 0 \leq \sum_{k=1}^{N} \mu_i(x_k) \leq N, \forall i \right\}$$

Partición difusa

El objetivo del agrupamiento es dividir el conjunto de datos X en c clusters. Por el momento, asumimos que c es conocido, basado en el conocimiento previo. Las posibles particiones difusas se pueden ver como una generalización de la partición dura.

Una partición difusa del conjunto de datos X puede representarse mediante una matriz

$U = [\mu_{i,k}]$ de dimensión $c \times N$.

Donde $\mu_{i,k}$ denota el grado de pertenencia que la k-ésima observación pertenece al c-ésimo cluster $(1 \leq k \leq N, 1 \leq i \leq c)$. Por lo tanto, la i-ésima fila de U contiene valores de la funcion de pertenencia del i-esimo conjunto difuso de X.

La matriz U es llamada matriz de partición difusa. Las condiciones para una matriz difusa están dadas por:

$$\mu_{i,k} \in [0,1], 1 \leq k \leq N, 1 \leq i \leq c, \tag{1.7}$$

$$\sum_{i=1}^{c} \mu_{i,k} = 1, 1 \leq k \leq N, \tag{1.8}$$

$$0 < \sum_{k=1}^{N} \mu_{i,k} < N, 1 \leq i \leq c \tag{1.9}$$

La segunda condición restringe la suma de cada columna a 1, y por lo tanto el número total de miembros de cada x_k en X es igual a 1. La distribución de pertenencias entre los c subconjuntos difusos no está restringida.

Definición 3 ***(Espacio de partición difusa)***

Sea $X = [x_1, x_2, \cdots, x_N]$ un conjunto finito y sea $2 \leq c < N$ un número entero. El espacio de partición difusa para X es el conjunto:

$$M_{fc} = \left\{ U \in \mathbb{R}^{c \times N} \mid \mu_{i,k} \in [0,1], \forall i,k; \sum_{i=1}^{c} \mu_{i,k} = 1, \forall k; 0 < \sum_{k=1}^{N} \mu_{i,k} < N, \forall i \right\} \tag{1.10}$$

Ejemplo: Sea $X = \{x_1 = melocoton, x_2 = fresa, x_3 = ciruela\}$ conjunto de datos. En vista de las restricciones para $U \in M_{fc}$ hay infinitas 2-particiones difusas. Una partición difusa típica que muestra bastante bien la utilidad de ésta integración es la siguiente:

$$\begin{matrix} x_1 & x_2 & x_3 \end{matrix} \tag{1.11}$$

$$\mathbf{U} = \begin{bmatrix} 0{,}91 & 0{,}58 & 0{,}13 \\ 0{,}09 & 0{,}42 & 0{,}87 \end{bmatrix} \tag{1.12}$$

Los valores de pertenencia en (1.12) indican que $x_1 = melocoton$ y $x_3 = ciruela$, tienen altas afinidades para diferentes cluster, mientras que $x_2 = fresa$, tiene caracteristicas que exigen una pertenencia parcial relativamente mayor en estos dos grupos difusos, la situación que anticipamos apartir de una idea intuitiva de la relacion real de los miembros de los datos. En otras palabras M_{fc} tiene un potencial significativamente mas alto que el caso de partición dura para modelar la realidades fisicas del conjunto de datos X. Surge la pregunta que si tenemos una base de datos real pueden particiones como (1.12) generarse apartir de éstos datos. la respuesta es en efecto si!, se discutirá en las siguintes secciones algoritmos que hacen esto.

La Función Difusa c-Means

Definición 4 ***(Función Difusa c-Means)***

Sea $J : M_{fc} \times \mathbb{R}^{c \times N} \to \mathbb{R}^{+}$

$$J(X; U, V) = \sum_{k=1}^{N} \sum_{i=1}^{c} (\mu_{i,k})^m d_{i,k}^2 \tag{1.13}$$

Donde

$$U = [\mu_{i,k}] \in M_{fc} \tag{1.14}$$

Es una matriz de partición difusa de X*,*

$$V = [v_1, v_2, \cdots, v_c] \in \mathbb{R}^{c \times N}, v_i \in \mathbb{R}^n \tag{1.15}$$

Es la matriz de prototipos de grupos (centros), que deben ser determinados,

$$d_{i,k}^2 = \|x_k - v_i\|^2, \tag{1.16}$$

$\|.\|$ *Es cualquier producto interno inducido por la norma en* $\mathbb{R}^n$ *y* $m \in [1, \infty)$ *es el exponente de ponderación.*

Algoritmo Difuso c-Means

La minimización de la función c-Means (1.13) representa un problema de optimización no lineal que se puede resolver utilizando una variedad de métodos disponibles, para entrar en detalle debemos tener en cuenta el siguiente resultado:

Teorema 1 *Asuma* $\|.\|$ *inducida por un producto interno. Fijamos* $m \in (1, \infty)$ *Sea* X *tenga al menos* $c < n$ *puntos distintos y definimos* $\forall k$ *el conjunto*

$$I_k = \{i \mid 1 \leq i \leq c; d_{i,k} = \|\mathbf{x}_k - \mathbf{v}_i\| = 0\}$$

$$\bar{I}_k = \{1, 2, \cdots, c\} - I_k$$

Entonces $(U, \mathbf{v}) \in M_{fc} \times \mathbb{R}^{c \times n}$ *Es un minimo global para J solo si*

$$I_k = \emptyset \Rightarrow \mu_{i,k} = \frac{1}{\sum_{j=1}^{c} \left(d_{i,k}/d_{j,k}\right)^{2/(m-1)}} \tag{1.17}$$

$$I_k \neq \emptyset \Rightarrow \mu_{i,k} = 0 \forall i \in \bar{I}_k, \sum_{i \in I_k} \mu_{i,k} = 1 \tag{1.18}$$

$$\mathbf{v}_i = \frac{\sum_{k=1}^{n} (\mu_{i,k})^m \mathbf{x}_k}{\sum_{k=1}^{n} (\mu_{i,k})^m}, \forall i \tag{1.19}$$

Análisis: El resultado (1.17) se obtiene fijando $\mathbf{v} \in \mathbb{R}^{N \times c}$ aplicando multiplicadores de Lagrange a las variables $\mu_{i,k}$. Hay un truco en la técnica, relajamos U permitiendo que esté en M_{fco}, de modo que la minimización pueda realizarse término a término en las columnas de U. Las soluciones resultantes siempre se cumplen en M_{fc}.

Fijamos $\mathbf{v} \in \mathbb{R}^{n \times c}$ y definimos $g(U) = J(U, \mathbf{v})$ para algun $U \in M_{fco}$. Dado U es degenerado, sus columnas son independientes, y por lo tanto:

$$\min_{U \in M_{fco}} \{g(U)\} = \min_{U \in M_{fco}} \left\{ \sum_{k=1}^{n} \sum_{i=1}^{c} (\mu_{i,k})^m (d_{i,k})^2 \right\} = \sum_{k=1}^{n} \left[\min_{\boldsymbol{\mu}_k \in conv(B_c)} \left\{ \sum_{i=1}^{c} (\mu_{i,k})^m (d_{i,k})^2 \right\} \right] \tag{1.20}$$

donde: $conv(B_c) = \{\boldsymbol{\mu}_k \in \mathbb{R}^c \mid \sum_{i=1}^{c} \mu_{i,k} = 1; \mu_{i,k} \geq 0\}$

La solución de (1.20) es efectuada con los multipicadores de Lagrange, para cada término, sea

$$g_k(\boldsymbol{\mu}_k) = \sum_{i=1}^{c} (\mu_{i,k})^m (d_{i,k})^2$$

y sea su Lagrangiano

$$F_k(\lambda, \boldsymbol{\mu}_k) = \sum_{i=1}^{c} (\mu_{i,k})^m (d_{i,k})^2 - \lambda \left(\sum_{i=1}^{c} \mu_{i,k} - 1 \right)$$

$(\lambda, \boldsymbol{\mu}_k)$ es estacionario para F_k solo si $\nabla_{\lambda, \boldsymbol{\mu}_k} \mathrm{F}_k(\lambda, \boldsymbol{\mu}_k) = 0$ Igualando el gradiente a cero:

$$\frac{\partial \mathbf{F}_k}{\partial \lambda}(\lambda, \boldsymbol{\mu}_k) = \left(\sum_{i=1}^{c} \mu_{i,k} - 1 \right) = 0 \tag{1.21}$$

$$\frac{\partial \mathbf{F}_k}{\partial \mu_{s,t}}(\lambda, \boldsymbol{\mu}_k) = \left[m(\mu_{s,t})^{m-1} (d_{s,t})^2 - \lambda \right] = 0 \tag{1.22}$$

De esto,

$$\mu_{st} = \left[\frac{\lambda}{m(d_{st})^2} \right]^{\frac{1}{m-1}} \tag{1.23}$$

Usando (1.21)

$$\sum_{j=1}^{c} \mu_{jt} = \sum_{j=1}^{c} \left(\frac{\lambda}{m} \right)^{\frac{1}{m-1}} \left[\frac{1}{(d_{jt})^2} \right]^{\frac{1}{m-1}} = \left(\frac{\lambda}{m} \right)^{\frac{1}{m-1}} \left\{ \sum_{j=1}^{c} \left[\frac{1}{(d_{jt})^2} \right]^{\frac{1}{m-1}} \right\} = 1$$

Así,

$$\left(\frac{\lambda}{m} \right)^{\frac{1}{m-1}} = \frac{1}{\sum_{j=1}^{c} \left(\frac{1}{(d_{jt})^2} \right)^{\frac{1}{m-1}}}$$

Volviendo a (1.23),

$$\mu_{st} = \left\{ \frac{1}{\sum_{j=1}^{c} \left[\frac{1}{(d_{jt})^2} \right]^{\frac{1}{m-1}}} \right\} \left[\frac{1}{(d_{st})^2} \right]^{\frac{1}{m-1}}$$

$$\mu_{st} = \frac{1}{\sum_{j=1}^{c} \left(d_{s,t} / d_{j,t} \right)^{2/(m-1)}}$$

En este punto, existe una de dos posibilidades: si $I_t = \emptyset$, entonces (1.17) sigue para la columna t; si $I_t \neq \emptyset$, entonces, eligiendo $\{\mu_{st}\}$ como en (1.18) resulta en $g_t(\boldsymbol{\mu}_t) = 0$, porque los pesos no cero se colocan en distancias cero, mientras que las distancias positivas aumentarán $g_t(\boldsymbol{\mu}_t)$, minimalidad contradictoria.

Continuando de esta manera, se llega a n vectores $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_n$, que, tomados juntos como una matriz U, definen un punto estacionario para g. Si todos los $\boldsymbol{u}_k$ son computados por (1.17), $\mu_{ik} \in (0,1) \forall i,k$ y U está claramente en M_{fc} (Cada entrada es difusa!). Si ocurren singularidades, lo que requiere el uso de (1.18) para algunas columnas, U aún no se genera. Para ver esto, supongamos, por el contrario, que para la fila r, $\mu_{rk} = 0 \forall k$. Definir

$$\mathbf{v}_i^* = \left\{ \begin{array}{ll} \mathbf{v}_i, & 1 \leq i \leq c;\ i \neq r \\ \mathbf{x}_n, & i = r \end{array} \right\}$$

donde $\mathbf{x}_n \in \boldsymbol{X}$, $\mathbf{x}_n \neq \mathbf{v}_i$ para $i \neq r$. Tal vector existe porque $\boldsymbol{X}$ contiene c puntos distintos. Ahora

$$0 = \sum_{k=1}^{n} (\mu_{rk})^m (d_{rk})^2 = \sum_{k=1}^{n} (\mu_{rk})^m (d_{rk}^*)^2$$

donde $d_{rk}^* = \|\mathbf{x}_k - \mathbf{v}_r^*\| \ \forall r$. Entonces $(U, \mathbf{v}^*)$ es el mínimo global de J. Ya que $\mathbf{v}_r^* = \mathbf{x}_n$, el conjunto $I_n = \{r\}$ con $\mu_{rn} = 1$ por (1.18). Esta contradicción demuestra que $U \in M_{fc}$ cuando (1.17) es utilizado para construirlo.

Establecer (1.19). Fijo $U \in M_{fc}$ y el conjunto $h_m(\mathbf{v}) = J(U, \mathbf{v})$. La minimización de h_m no está restringida sobre $\mathbb{R}^{c \times p}$, por lo que tenemos

$$\begin{aligned} h_m(\mathbf{v}) &= \sum_{k=1}^{n} \sum_{i=1}^{c} (\mu_{ik})^m (d_{ik})^2 \\ &= \sum_{k=1}^{n} \sum_{i=1}^{c} (\mu_{ik})^m \langle \mathbf{x}_k - \mathbf{v}_i, \mathbf{x}_k - \mathbf{v}_i \rangle \end{aligned}$$

dónde $\langle \cdot, \cdot \rangle$ es el producto interior que induce la norma. Entonces para cada i, es necesario que las derivadas direccionales $h_m'(\mathbf{v}_i; \mathbf{w})$ desaparezcan para todos los

vectores unitarios $\mathbf{w} \in \mathbb{R}^c$,es decir,$h'_m(\mathbf{v}_i; \mathbf{w}) = 0 \ \ \forall \mathbf{w}$:

$$
\begin{aligned}
h'_m(\mathbf{v}_i; \mathbf{w}) &= \sum_{k=1}^{n} (\mu_{i,k})^m \frac{d}{dt}(\langle \mathbf{x}_k - \mathbf{v}_i - t\mathbf{w}, \mathbf{x}_k - \mathbf{v}_i - t\mathbf{w} \rangle)|_{t=0} \\
&= \sum_{k=1}^{n} (\mu_{i,k})^m \left(\frac{d}{dt}(\mathbf{x}_k - \mathbf{v}_i - t\mathbf{w}) \cdot (\mathbf{x}_k - \mathbf{v}_i - t\mathbf{w}) + (\mathbf{x}_k - \mathbf{v}_i - t\mathbf{w}) \cdot \frac{d}{dt}(\mathbf{x}_k - \mathbf{v}_i - t\mathbf{w}) \right) |_{t=0} \\
&= \sum_{k=1}^{n} (\mu_{i,k})^m \left(-\mathbf{w} \cdot (\mathbf{x}_k - \mathbf{v}_i - t\mathbf{w}) - \mathbf{w} \cdot (\mathbf{x}_k - \mathbf{v}_i - t\mathbf{w}) \right) |_{t=0} \\
&= -2 \sum_{k=1}^{n} (\mu_{i,k})^m \left(\mathbf{w} \cdot (\mathbf{x}_k - \mathbf{v}_i - t\mathbf{w}) \right) |_{t=0} \\
&= -2 \sum_{k=1}^{n} (\mu_{i,k})^m \left(\mathbf{w} \cdot (\mathbf{x}_k - \mathbf{v}_i) \right) \\
&= -2 \left[\sum_{k=1}^{n} (\mu_{i,k})^m \langle \mathbf{x}_k - \mathbf{v}_i, \mathbf{w} \rangle \right] = 0 \ \ \forall \mathbf{w} \\
&\Leftrightarrow \langle \sum_{k=1}^{n} (\mu_{i,k})^m (\mathbf{x}_k - \mathbf{v}_i), \mathbf{w} \rangle = 0 \ \ \forall \mathbf{w} \\
&\Leftrightarrow \sum_{k=1}^{n} (\mu_{i,k})^m (\mathbf{x}_k - \mathbf{v}_i) = \mathbf{0}
\end{aligned}
$$

de lo que (1.19) sigue.

Tenga en cuenta que las condiciones del teorema se cumplen para cualquier norma inducida por el producto interno métrico. En particular, cualquier matriz definida positiva A de orden $n \times n$ induce una norma a través del producto interno ponderado

$$\langle \mathbf{x}, \mathbf{y} \rangle_A = \mathbf{x}^T A \mathbf{y} = \sum_{i=1}^{n} \sum_{j=1}^{n} \mathbf{x}_i^- a_{ij} y_j \tag{1.24}$$

$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}$. Relativo a esta clase especial de normas J puede escribirse como:

$$J(U, \mathbf{v}, A) = \sum_{k=1}^{N} \sum_{i=1}^{c} (u_{ik})^m \|\mathbf{x}_k - \mathbf{v}_i\|_A^2 \tag{1.25}$$

dónde

$$(d_{ik})^2 = \|\mathbf{x}_k - \mathbf{v}_i\|_A^2 = \langle \mathbf{x}_k - \mathbf{v}_i, \mathbf{x}_k - \mathbf{v}_i \rangle_A = (\mathbf{x}_k - \mathbf{v}_i)^T A (\mathbf{x}_k - \mathbf{v}_i)$$

Esta forma enfatiza la dependencia de J en la matriz A que define la norma para $\mathbb{R}^n$. Hay dos razones para hacerlo: bajo ciertas condiciones especiales, A puede

incluirse como una variable teórica para la optimización, como en la modificación de Gustafson y Kessel. La dependencia de J en A se suprimirá a continuación, a menos que la situación lo justifique.

El algoritmo de cluster difuso $c-$means es simplemente una iteración de Picard a través de condiciones necesarias del teorema.

$$\bar{J}(X;U,V,\lambda) = \sum_{i=1}^{c}\sum_{k=1}^{N}(\mu_{i,k})^m D_{i,kA}^2 + \sum_{k=1}^{N}\lambda_k\left(\sum_{i=1}^{c}\mu_{i,k} - 1\right) \tag{1.26}$$

y estableciendo los gradientes de $\bar{J}$ con respecto a U, V y λ a cero. Si $D_{i,kA}^2 > 0, \forall i,k$ y $m > 1$, entonces $(U,V) \in M_{fc} \times \mathbb{R}^{N\times c}$ puede minimizar (1.25) solo si

$$\mu_{i,k} = \frac{1}{\sum_{j=1}^{c}(D_{i,kA}/D_{j,kA})^{2/m-1}}, 1 \le i \le c, 1 \le k \le N, \tag{1.27}$$

y

$$\mathbf{v}_i = \frac{\sum_{k=1}^{N}(\mu_{i,k})^m \mathbf{x}_k}{\sum_{k=1}^{N}(\mu_{i,k})^m}, 1 \le i \le c. \tag{1.28}$$

Algoritmo Difuso c-Means

Dado el conjunto de datos X, elija el número de agrupaciones $1 < c < N$, el exponente de ponderación $m > 1$, la tolerancia de terminación $\epsilon > 0$ y la matriz de normas inducidas a la matriz A. Inicialice la matriz de partición al azar de modo que $U^{(0)} \in M_{fc}$

Repetir para $l = 1, 2, \cdots$

Paso 1 Calcular los prototipos de clusters (medias):

$$v_i^{(l)} = \frac{\sum_{k=1}^{N}(\mu_{i,k}^{(l-1)})^m x_k}{\sum_{k=1}^{N}(\mu_{i,k}^{(l-1)})^m}, 1 \le i \le c. \tag{1.29}$$

Paso 2 Calcular las distancias:

$$D_{i,kA}^2 = (x_k - v_i)^T A(x_k - v_i), 1 \le i \le c, 1 \le k \le N. \tag{1.30}$$

Paso 3 Actualizar la matriz de particiones :

$$u_{i,k}^{(l)} = \frac{1}{\sum_{j=1}^{c}(D_{i,kA}/D_{j,kA})^{2/m-1}}. \tag{1.31}$$

Mientras

$$\|U^{(l)} - U^{(l-1)}\| < \epsilon$$

Si los datos están normalizados (es decir, todas las características tienen media cero y varianza uno), los resultados del agrupamiento cambiarán, los agrupamientos tienen una forma naturalmente circular, pero los centros del agrupamiento son diferentes, están más bien ubicados uno por encima del otro por los datos originales (también con diferentes estados iniciales). Por consiguiente, el algoritmo difuso C-means es sensible a la escala (normalización) de los datos.

1.1.5. Algoritmo de Gustafson-Kessel (GK)

Gustafson y Kessel extendieron el algoritmo difuso c-Means estándar mediante el uso de una norma de distancias adaptativa a fin de detectar grupos de diferente forma geométrica en un conjunto de datos. Cada grupo tiene su propia matriz A_i que induce las normas, lo que produce la siguiente norma del producto interno.

$$D_{i,kA}^2 = (x_k - v_i)^T A_i (x_k - v_i), 1 \leq i \leq c, 1 \leq k \leq N. \tag{1.32}$$

Las matrices A_i se utilizan como variables de optimización en la función c-means, lo que permite que cada grupo adapte la norma de distancia a la estructura topológica local de los datos. Sea A una c-tupla de las matrices que inducen la norma:

$A = (A_1, A_2, \cdots, A_c)$.

La función objetivo del algoritmo GK se define por:

$$J(X; U, V, A) = \sum_{i=1}^{c}\sum_{k=1}^{N}(\mu_{i,k})^m D_{i,kA_i}^2. \tag{1.33}$$

Para A fija, las condiciones (1.7), (1.8) y (1.9) pueden ser directamente aplicadas. Sin embargo la función objetivo (1.33) no se puede minimizar directamente con respecto a A_i, ya que es lineal en A_i. Esto significa que J puede hacerse tan pequeño como se desee simplemente haciendo que A_i sea menos definida positiva.

Para obtener una solución factible, A_i debe estar restringida de alguna manera. La forma habitual de lograr esto es restringir el determinante de A_i. Permitir que la matriz A_i varié con su determinante fijo corresponde a optimizar la forma de los grupos mientras su volumen permanece constante:

$$det(A_i) = \rho_i, \rho > 0 \tag{1.34}$$

Donde ρ_i es fijo para cada cluster.

Usando el método de multiplicadores de Lagrange se obtiene la siguiente expresión para A_i:

$$A_i = [\rho_i det(F_i)]^{\frac{1}{n}} F_i^{-1}, \tag{1.35}$$

Donde F_i es la matriz de covarianza difusa del i-ésimo cluster definido por:

$$F_i = \frac{\sum_{k=1}^{N} (\mu_{i,k})^m (x_k - v_i)(x_k - v_i)^T}{\sum_{k=1}^{N} (\mu_{i,k})^m} \tag{1.36}$$

Note que la sustitución de (1.35) y (1.36) en (1.32) proporciona una norma de distancia de Mahalanobis generalizada al cuadrado entre x_k y la media del cluster v_i, donde la covarianza es ponderada por los grados de pertenencia en U.

La descripción formal del algoritmo de clustering GK se presenta a continuación:

Algoritmo Gustafson-Kessel

Dado el conjunto de datos X, elija el número de agrupaciones $1 < c < N$, el exponente de ponderación $m > 1$, la tolerancia de terminación $\epsilon > 0$ y la matriz de normas inducidas a la matriz A.
Inicialice la matriz de partición al azar de modo que $U^{(0)} \in M_{fc}$

Repetir para $l = 1, 2, \cdots$

Paso 1 Calcular los centros de los clusters:

$$\mathrm{v}_i^{(l)} = \frac{\sum_{k=1}^{N} (\mu_{i,k}^{(l-1)})^m \mathrm{x}_k}{\sum_{k=1}^{N} (\mu_{i,k}^{(l-1)})^m}, 1 \leq i \leq c. \tag{1.37}$$

Paso 2 Calcular las matrices de covarianza de los clusters:

$$\mathrm{F}_i^{(l)} = \frac{\sum_{k=1}^{N} (\mu_{i,k}^{(l-1)})^m (\mathrm{x}_k - \mathrm{v}_i^{(1)})(\mathrm{x}_k - \mathrm{v}_i^{(1)})^T}{\sum_{k=1}^{N} (\mu_{i,k}^{(l-1)})^m}, 1 \leq i \leq c. \tag{1.38}$$

Paso 3 Calcular las distancias:

$$D^2_{i,k\mathrm{A}_i}(\mathrm{x}_k, \mathrm{v}_i) = (\mathrm{x}_k - \mathrm{v}_i)^T [(\rho_i det(\mathrm{F}_i))^{\frac{1}{n}} F_i^{-1}](\mathrm{x}_k - \mathrm{v}_i). \tag{1.39}$$

Paso 4 Actualizar la matriz de particiones:

$$u_{i,k}^{(l)} = \frac{1}{\sum_{j=1}^{c} (D^2_{i,k\mathrm{A}_i}(\mathrm{x}_k, \mathrm{v}_i) / D^2_{j,k\mathrm{A}_j}(\mathrm{x}_k, \mathrm{v}_j))^{2/m-1}}, 1 \leq i \leq c, 1 \leq k \leq N. \tag{1.40}$$

Mientras

$$\|\mathrm{U}^{(l)} - \mathrm{U}^{(l-1)}\| < \epsilon.$$

1.1.6. Algoritmo de Agrupamiento de Gath-Geva

El algoritmo de agrupamiento de estimaciones de probabilidad máxima difusa emplea una distancia norma basada en las estimaciones difusas de máxima

verosimilitud propuestas por Bezdek y Dunn [4].

$$D_{i,k}(x_k, v_i) = \frac{(2\pi)^{\frac{n}{2}}\sqrt{det(F_i)}}{\alpha_i} exp\left(\frac{1}{2}(x_k - v_i)^T F_i^{-1}(x_k - v_i)\right) \tag{1.41}$$

Tenga en cuenta que, al contrario del algoritmo GK ésta norma de distancia implica un término exponencial y por lo tanto, disminuye más rápido que la norma del poducto interno. F_i denota la matriz de covarianza difusa del *i-simo* grupo, similar al algoritmo GK, dado por (1.36). El α_i es la probabilidad previa de seleccionar el grupo i dada por:

$$\alpha_i = \frac{1}{N}\sum_{k=1}^{N} \mu_{i,k} \tag{1.42}$$

Los grados de pertenencia $\mu_{i,k}$ se interpretan como la probabilidades posteriores de seleccionar el i-ésimo clúster dado los puntos de datos x_k.

Gath y Geva informaron que el algoritmo de agrupamiento de estimaciones de probabilidad máxima difusa es capaz de detectar clústers de diferentes formas, tamaños y densidades. Ésto se debe a que la matriz de covarianza del clúster se utiliza junto con una distancia exponencial, y los grupos no están restringidos en volumen. Sin embargo, este algoritmo es menos robusto en el sentido que necesita una buena inicialización, ya que debido a la norma de distancia exponencial, converge a un óptimo local cercano.

El mínimo de 1.13 es buscado por el método de optimización alterna(AO)(algoritmo de agrupación Gath-Geva) dado en el algoritmo Gath-Geva.

Algoritmo Gath-Geva

Dado el conjunto de datos X, especificamos c, elija el exponente de ponderación $m > 1$, una tolerancia de terminación $\epsilon > 0$. Inicializar la matriz de partición de modo que contengan (1.7),(1.8) y (1.9).

Repetir para $l = 1, 2, \cdots$

Paso 1 Calcular los centros de los clusters:

$$\mathrm{v}_i^{(l)} = \frac{\sum_{k=1}^{N} (\mu_{i,k}^{(l-1)})^m \mathrm{x}_k}{\sum_{k=1}^{N} (\mu_{i,k}^{(l-1)})^m}, 1 \leq i \leq c. \tag{1.43}$$

Paso 2 Calcular las medidas de distancia $D_{i,k}^2$.
La distancia al prototipo se calcula en función de las matrices de covarianza difusa de los clusters:

$$\mathrm{F}_i^{(l)} = \frac{\sum_{k=1}^{N} (\mu_{i,k}^{(l-1)})^m (\mathrm{x}_k - \mathrm{v}_i^{(l)})(\mathrm{x}_k - \mathrm{v}_i^{(l)})^T}{\sum_{k=1}^{N} (\mu_{i,k}^{(l-1)})^m}, 1 \leq i \leq c. \tag{1.44}$$

La función de distancia se elige como:

$$D_{i,k}(x_k, v_i) = \frac{(2\pi)^{\frac{n}{2}} \sqrt{det(F_i)}}{\alpha_i} exp\left(\frac{1}{2}(x_k - v_i^l)^T F_i^{-1}(x_k - v_i^l)\right) \tag{1.45}$$

Con la probabilidad a prori $\alpha_i = \frac{1}{N} \sum_{k=1}^{N} \mu_{i,k}$

Paso 3 Actualizar la matriz de particiones:

$$u_{i,k}^{(l)} = \frac{1}{\sum_{j=1}^{c} (D_{i,k}^2(\mathrm{x}_k, \mathrm{v}_i) / D_{j,k}^2(\mathrm{x}_k, \mathrm{v}_j))^{2/m-1}}, 1 \leq i \leq c, 1 \leq k \leq N. \tag{1.46}$$

Mientras

$$\|\mathrm{U}^{(l)} - \mathrm{U}^{(l-1)}\| < \epsilon.$$

1.1.7. Visualización de resultados del agrupamiento.

Desde antes en los problemas prácticos de la minería de datos, los datos con alta dimensionalidad son aglomerados, los grupos resultantes son objetos geométricos de alta dimensionalidad que son difíciles de analizar e interpretar. El agrupamiento siempre acomoda los grupos de datos, aun si la estructura del grupo no es adecuada para el problema. Para analizar la adecuación de los grupos prototipo y el número de los grupos, las medidas de validez de grupos son utilizadas.

Sin embargo, aunque las medidas de validación reduzcan la evaluación global a un solo número, no pueden evitarse una cierta pérdida de información.

Una representación gráfica de baja dimensionalidad de los grupos podría ser mucho más informativa que un simple valor de la validez de grupos porque uno puede aglomerar por medio de la vista y validar cualitativamente extrayendo conclusiones de los algoritmos de agrupamiento. Este capítulo presenta la visualización de datos de alta dimensionalidad general, y presenta dos nuevos métodos para la visualización de resultados de agrupamiento difuso.

1.1.8. Análisis de componentes principales.

PCA toma un conjunto de datos $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, x_n]^T$ donde $\mathbf{x}_k[x_{1,k}, x_{2,k}, \ldots, x_{n,k}]^T$ es la k-ésima muestra o dato puntual en una base ortonormal dada en $\mathbf{R}^m$ y encontramos una nueva base ortonormal $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_n]$ con $\mathbf{u}_i = [u_{1,i}, \ldots, u_{n,i}]^T$ con sus ejes ordenados.

Esta base nueva es rotada de tal manera que el primer eje está orientado a lo largo de la dirección en la cual los datos tienen su varianza máxima. El segundo eje está orientado a lo largo de la dirección de la varianza máxima en los datos, ortogonal al primer eje. De modo semejante, los subsiguientes ejes están orientados para dar explicación lo más posible de la variabilidad en los datos, sujetos a la restricción que deben ser ortogonales a los ejes precedentes.

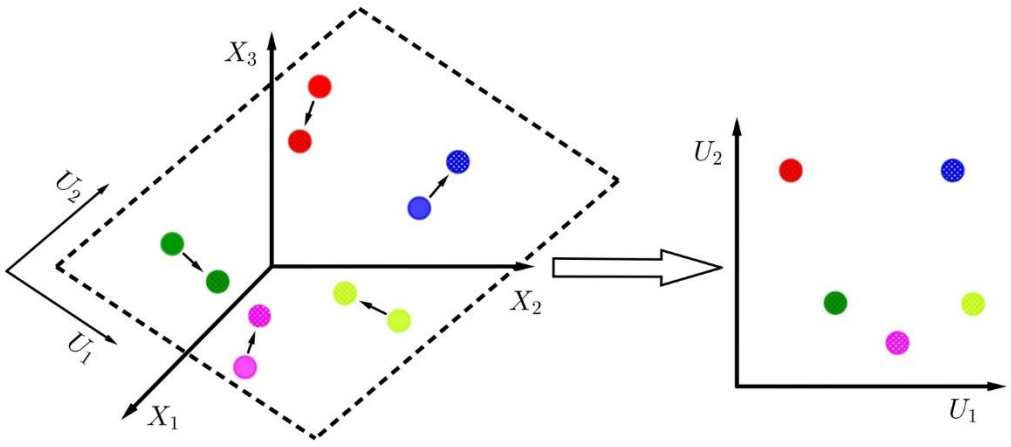


Figura 1.1: Ilustración para PCA (Análisis Componentes Principales).

Consecuentemente, estos ejes tienen asociado 'índices' decrecientes λ_i, $i = 1, 2, \ldots, n$, correspondiente a la varianza del conjunto de datos proyectados sobre los ejes.
Las componentes principales son los nuevos vectores de la base, ordenados por sus varianzas correspondientes. El vector con la varianza más grande corresponde a la primera componente principal. Proyectando el conjunto de datos original en las q primeras componentes principales, con $q < n$ un nuevo conjunto de datos con dimensionalidad inferior (principalmente 2 o 3 con propósito de visualización) puede ser obtenido. Si los componentes principales son primeramente escaladas por las varianzas inversas correspondientes, entonces las variables del nuevo conjunto de datos tendrán una variación de unidades, un procedimiento conocido como blanqueado o nivelación.
La forma tradicional de calcular las componentes principales es calcular la matríz de covarianzas muestral del conjunto de datos,

$$\mathbf{F} = \frac{1}{N-1}\sum_{k=1}^{N}(\mathbf{x}_k - \bar{\mathbf{x}})(\mathbf{x}_k - \bar{\mathbf{x}})^T,\ \bar{\mathbf{x}} = \frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_k \tag{1.47}$$

Y luego encontramos la eigen estructura:

$$\mathbf{FU} = \mathbf{U\Lambda} \tag{1.48}$$

$\mathbf{U}$ es una matriz de $n \times n$ que tiene los vectores propios de longitud la unidad en sus columnas y Λ es una matriz diagonal con los correspondientes valores propios λ_i,$i = 1, 2, \ldots, n$ a lo largo de la diagonal.
La varianza del conjunto de datos es igual a

$$\sigma^2 = \sum_{i=1}^{n} \lambda_i, \tag{1.49}$$

Por lo tanto, si sólo los primeros pocos (en su mayor parte 2) valores propios más grandes se usa para visualizar los datos multidimensionales originales, entonces la suma de los valores propios restantes se pierde.
Los vectores propios son las componentes principales y los valores propios son las correspondientes varianzas. En este caso $\mathbf{y}_k = \mathbf{U}^T\mathbf{x}_k$ es la representación de los k muestras en la nueva base. (Con q vectores) y es una aproximación del espacio original $\hat{\mathbf{x}}_k = \mathbf{U}\mathbf{U}^T\mathbf{x}_k$.
Una propiedad importante de los componentes principales es que constituyen el único conjunto de vectores que minimiza el error de reconstitución

$$Q = \sum_{k=1}^{N} (\mathbf{x}_k - \hat{\mathbf{x}}_k)^T(\mathbf{x}_k - \hat{\mathbf{x}}_k) = \sum_{k=1}^{N} ||\mathbf{x}_k - \mathbf{U}\mathbf{U}^T\mathbf{x}_k||^2 = \sum_{k=1}^{N} \mathbf{x}_k^T(I - \mathbf{U}\mathbf{U}^T)\mathbf{x}_k. \tag{1.50}$$

Q es la suma de las distancias al cuadrado entre los puntos de datos y sus proyecciones sobre las q componentes principales, sumadas sobre el conjunto de datos. Así, es una función decreciente de q, igual a cero cuando $q = n$. bajo esta formulación, PCA es conocida como la transformada de Karhunen-Loeve, y sugiere una vía alternativa para encontrar las componentes principales, minimizando (1.50). Este acercamiento ha formado la base para extensiones no lineales. El análisis de la distribución de los datos proyectados es también informativo. La medida T^2 de Hotelling se usa a menudo para calcular la distancia de los datos trazados desde

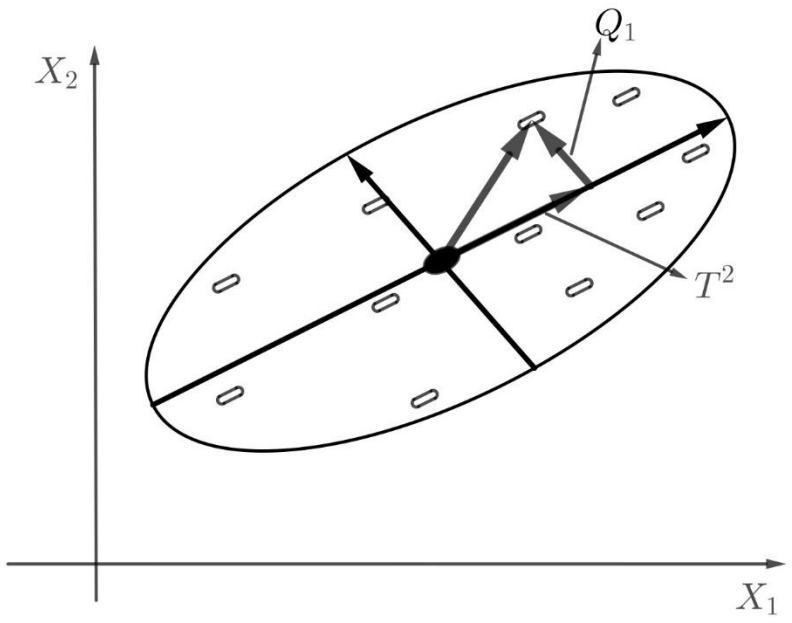


Figura 1.2: Las distancias medidas basadas sobre el modelo PCA.

un sub-espacio lineal.

$$T^2 = \sum_{k=1}^{N} \mathbf{y}_k^T \mathbf{y}_k \tag{1.51}$$

La (Figura 1.2) ilustra estas medidas en caso de dos variables y una componente principal.

Estas medidas T^2 y Q sirven a menudo para monitoreo de sistemas multivariables y para la exploración de los errores y las causas de los errores. Estas medidas T^2 y Q sirven a menudo para monitoreo de sistemas multivariables y para la exploración de los errores y las causas de los errores.

Ejemplo 1 ***(La Visualización de datos planta Iris basados en el Análisis De Componentes Principales).***
El conjunto de datos que se analiza es la base de datos Iris que consiste en los datos de las tres especies de la flor Iris descritas en el ejemplo de aplicacón de Gustaffson-Kessel, cuyas variabes o caracteristicas que se toman son las mismas.

Al ejecutar el codigo en el leguaje R, se obtiene lo que son las desviaciones estándar y los eigenvectores correspondientes a cada componente principal.

```
> iris.pca
Standard deviations (1, .., p=4):
[1] 2.0562689 0.4926162 0.2796596 0.1543862

Rotation (n x k) = (4 x 4):
                     PC1         PC2         PC3         PC4
Sepal.Length  0.36138659 -0.65658877  0.58202985  0.3154872
Sepal.Width  -0.08452251 -0.73016143 -0.59791083 -0.3197231
Petal.Length  0.85667061  0.17337266 -0.07623608 -0.4798390
Petal.Width   0.35828920  0.07548102 -0.54583143  0.7536574
```

Ademas obtenemos la información siguiente:

> summary(iris.pca)

		PC1	PC2	PC3	PC4
Importance of components:	Standard deviation	2.0563	0.49262	0.2797	0.15439
	Proportion of Variance	0.9246	0.05307	0.0171	0.00521
	Cumulative Proportion	0.9246	0.97769	0.9948	1.00000

Podemos observar que con las primeras dos componentes principales se logra el 97,8 % de la variabilidad de los datos, lo cual es bastante aceptable, con lo cual podemos reducir la dimensión de 4 variables cuantitativas a solamente dos.
En la visualización de resultados pueden verse la representación de las primeras dos componentes principales. Se nota claramente los tres grupos que se forman. Puede notarse además la separación que existe entre los grupos. La especie Setosa tiene menos semejanzas con las otras dos especies de flores iris.

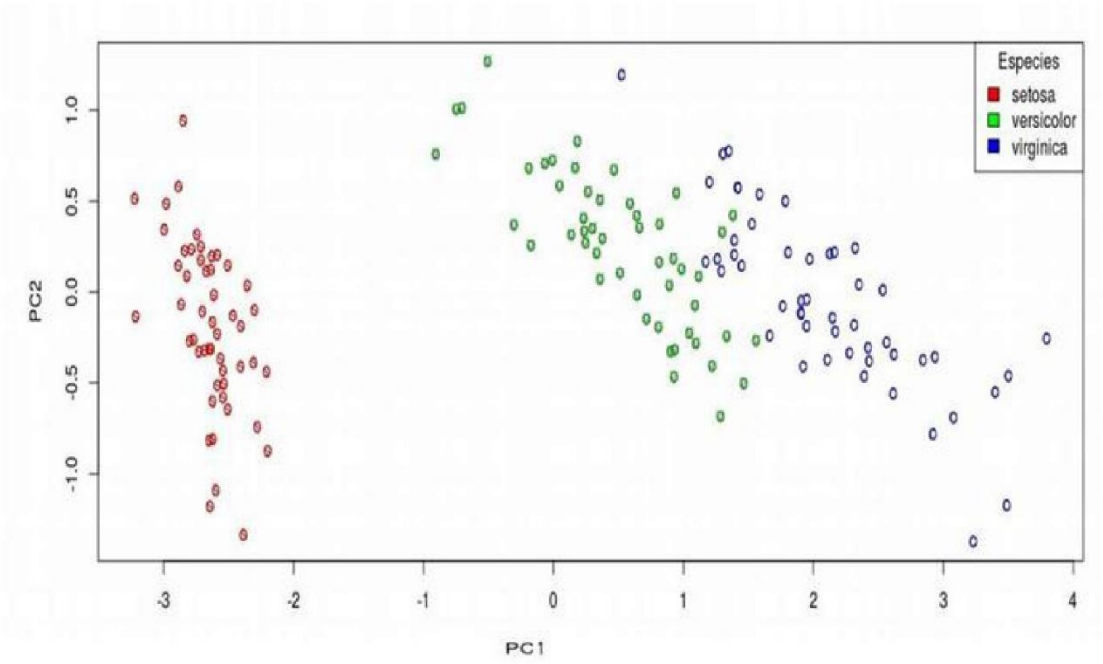


Figura 1.3: Datos de planta Iris: 3 especies

Ejemplo 2 ***(La Visualización de datos de Vino basados en el Análisis De Componentes Principales).***

Los datos de vino, que está disponible desde la Universidad de California, Irvine, por vía anónima ftp ftp.ics.uci.edu/pub/machine//machine-learning-databases, contiene el análisis químico de 178 vinos cultivados en la misma región en Italia pero derivativos de tres cultivos diferentes.

El problema es distinguir los tres tipos diferentes basados en 13 atributos continuos derivados del análisis químico: alcohólico, ácido Malic, Ceniza, Alcalinidad de la ceniza, Magnesio, fenoles Totales, Flavanoids, fenoles Nonflavanoids, Proanthocyaninsm, intensidad del color, Hue, OD280/OD315 de dilución del vino y Proline (Figura 1.4).

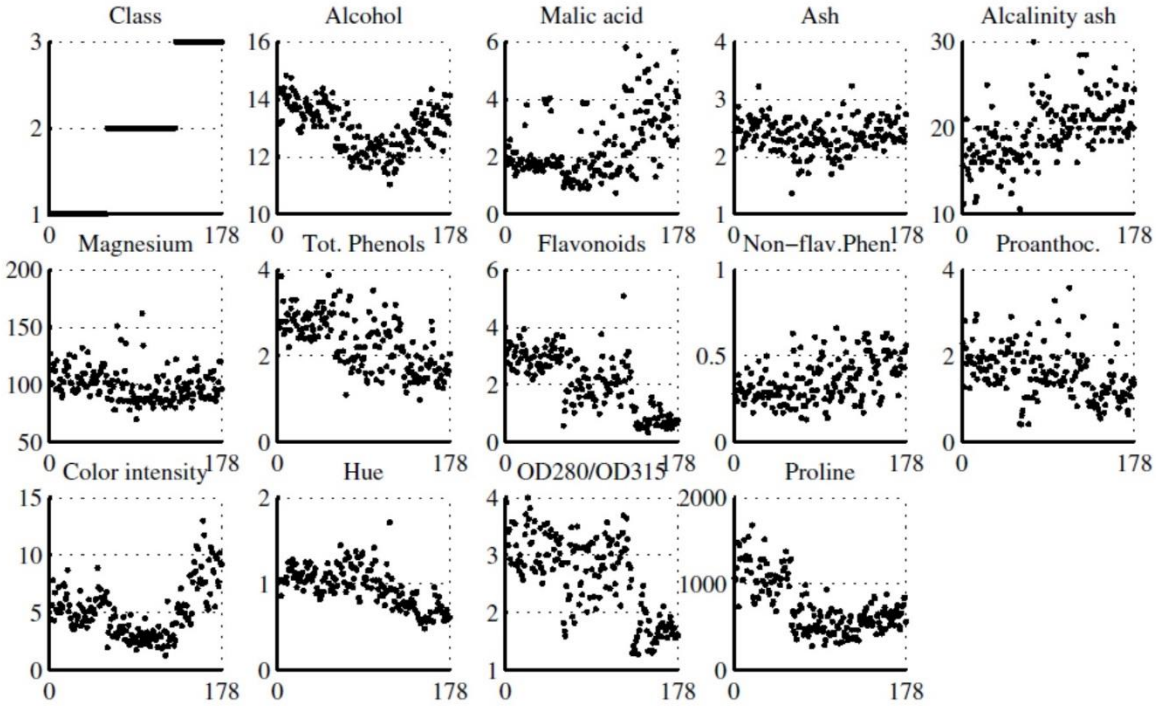


Figura 1.4: Datos de vino: 3 clases y 13 atributos.

El análisis ACP puede ser usado para visualización de las muestras en dos dimensiones. Los resultados pueden ser vistos en la (Figura 1.5).
Los tres tipos diferentes de vino se localizan en tres regiones adecuadas de separación del espacio del que se extendió a lo largo por las primeras dos componentes principales. Sin embargo, debería mencionarse que los primeros dos valores propios contengan sólo el 55 % de la variabilidad total, así es que el ACP puede resultar en falsos valores en caso del problema de clasificación.

Los valores propios y su suma acumulativa pueden ser vistos en la (Figura 1.6). La figura arriba mostrada es llamada gráfica de sedimentación que trama los valores propios ordenados según su contribución a la varianza de los datos.

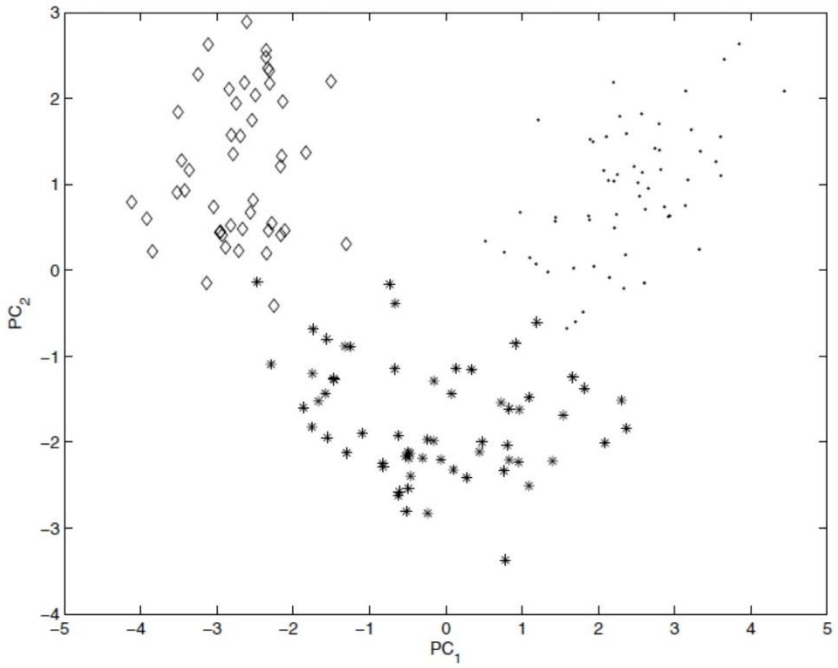


Figura 1.5: Análisis PCA de los datos de vino.

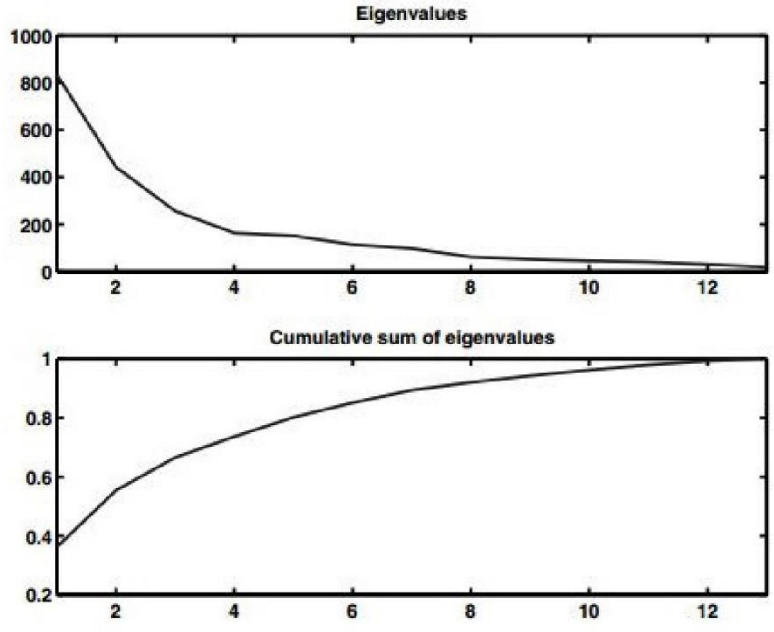


Figura 1.6: Gráfico de sedimentación de los datos de vino.

1.1.9. Mapeo de Sammon.

Mientras el ACP trata de conservar la variabilidad de los datos durante el mapeo, el mapeo de Sammon intenta conservar las distancias interpatrón. Con este propósito, Sammon definió la media de los errores al cuadrado entre las distancias en el espacio de dimensional alto y las distancias en el espacio proyectado de dimensional baja. Esta fórmula del error cuadrado es similar al criterio de estrés del escalamiento multidimensional.

El mapeo de Sammon es un procedimiento bien conocido para trazar un mapa de los datos de un espacio alto de dimensión n sobre un espacio inferior de dimensión q encontrando los N puntos en el espacio de datos de dimensión q, tal que la distancia entre los puntos $d_(i,j)^* = d^*(\mathbf{y}_i, \mathbf{y}_j)$ en el espacio de dimensión q aproximen las distancias correspondientes entre los puntos $d_(i,j) = d(\mathbf{x}_i, \mathbf{y}_j)$ en el espacio de dimensión n. Vea (Figura 1.7).

Esto es logrado al minimizar un criterio de error, llamado el estrés de Sammon, E:

$$E = \frac{1}{\lambda} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \frac{(d_{i,j} - d_{i,j}^*)^2}{d_{i,j}} \tag{1.52}$$

donde

$$\lambda = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d_{i,j}.$$

La minimización de E es un problema de optimización en N_q variables $y_{i,l}$, $i = 1, 2, \ldots, N$, $l = 1, 2, \ldots, q$ como $\mathbf{y}_i = [y_{i,1}, \ldots, y_{i,q}]^T$ Sammon aplicó el método de acantilado decente para minimizar esta función. Introduce la estimación de $y_{i,l}$ en la $t-$ésima iteración

$$y_{i,l}(t+1) = y_{i,l}(t) - \alpha \left[\frac{\frac{\partial E(t)}{\partial y_{i,l}(t)}}{\frac{\partial^2 E(t)}{\partial^2 y_{i,l}(t)}} \right] \tag{1.53}$$

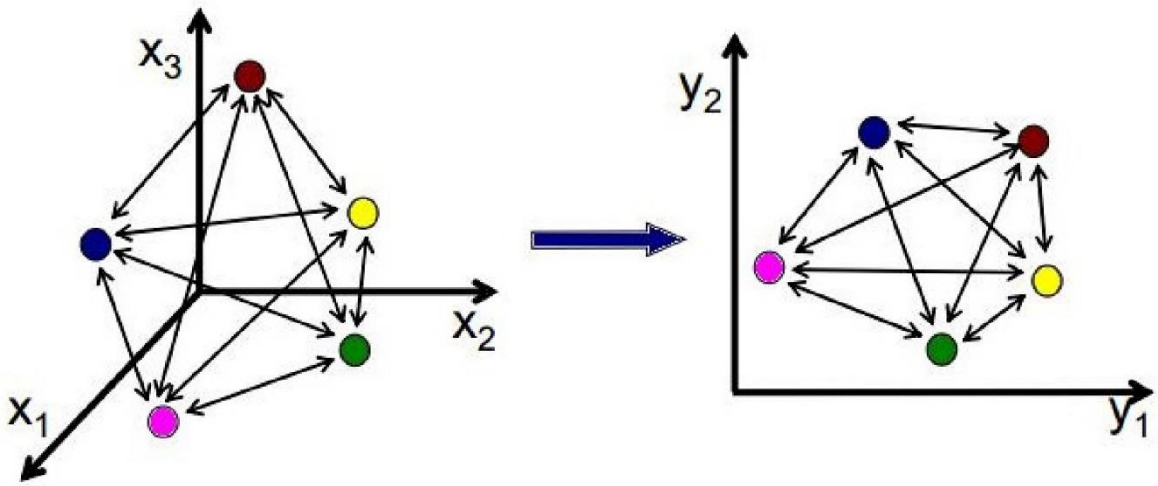


Figura 1.7: Ilustración para el mapeo de Sammon.

Donde α es una constante escalar no negativa (se recomienda $\alpha \simeq 0{,}3 - 0{,}4$), es decir, el tamaño de paso para la búsqueda del gradiente en la dirección de

$$\begin{aligned}
\frac{\partial E(t)}{\partial y_{i,l}(t)} &= -\frac{2}{\lambda} \sum_{k=1,k\neq i}^{N} \left[\frac{d_{k,i} - d^*_{k,i}}{d_{k,i} d^*_{k,i}} \right] (y_{i,l} - y_{k,l}) \\
\frac{\partial^2 E(t)}{\partial^2 y_{i,l}(t)} &= -\frac{2}{\lambda} \sum_{k=1,k\neq i}^{N} \frac{d_{k,i} - d^*_{k,i}}{d_{k,i} d^*_{k,i}} \left[d_{k,i} - d^*_{k,i} \right. \\
&\quad \left. - \left(\frac{(y_{i,l} - y_{k,l})^2}{d^*_{k,i}} \right) \left(1 + \frac{d_{k,i} - d^*_{k,i}}{d_{k,i}} \right) \right]
\end{aligned} \tag{1.54}$$

No es necesario mantener λ para una solución exitosa del problema de optimización, desde la minimización de $\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \frac{(d_{i,j} - d^*_{i,j})^2}{d_{i,j}}$ dará los mismos resultados. Cuando el método del gradiente descendente es aplicado para la búsqueda del estrés mínimo de Sammon, un mínimo local en la superficie de error puede ser cumplido. Por eso un número significativo de corridas con inicializaciones aleatorias diferentes puede ser necesario. Sin embargo, la inicialización de $\mathbf{y}$ puede basarse en información que es obtenida de los datos, como la primera y segunda normas de los vectores característicos o los ejes principales de la matriz de covarianzas de los datos

Ejemplo 3 ***(la Visualización de los datos de vino basados en mapeo de Sammon).*** *En la (Figura 1.8) pueden verse los resultados dados por el clásico mapeo de Sammon. Comparado a los resultados dados por el ACP (la Figura 1.5), puede determinarse que las clases no están tan distintas como se muestran por el ACP.*

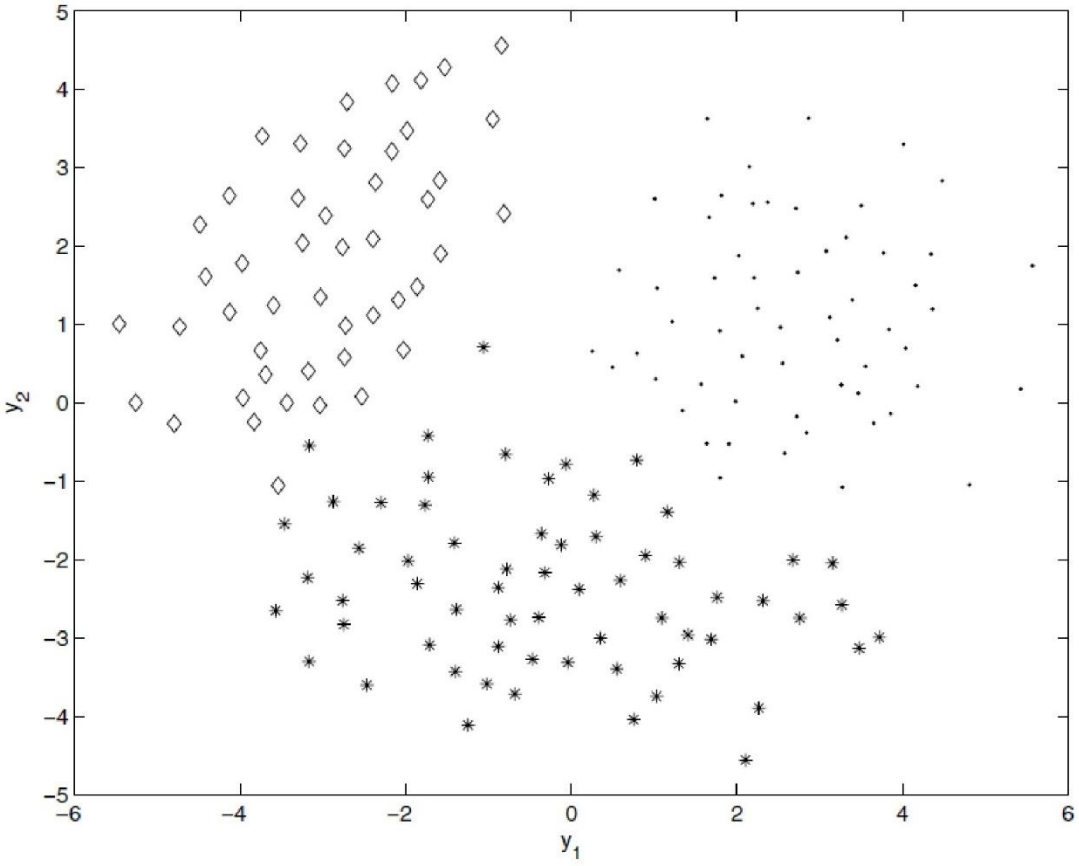


Figura 1.8: Datos del vino visualizado mediante el mapeo de Sammon.

1.1.10. Mapas Auto-organizados de Kohonen

Los mapas auto-organizados (SOM por sus siglas en inglés) es una nueva y efectiva herramienta para la visualización de datos de alta dimensionalidad. Implementa un mapeo ordenado de una distribución de alta dimensionalidad sobre una cuadrícula regular de baja dimensionalidad. Consecuentemente puede convertir relaciones complejas, no lineales y estadísticas entre datos de objetos de

alta dimensión en relaciones geométricas simples en un despliegue de baja dimensionalidad. Como es comprimir información conservando la topología y las relaciones métricas más importante de la información primaria de los objetos en el despliegue, también se puede razonar para producir algún tipo de abstracciones. Estos dos aspectos, la visualización y la abstracción, pueden ser utilizados en un número de vías de tareas complejas como análisis de proceso, percepción de máquina, control, y comunicación.

SOM realiza una conservación de la topología haciendo mapas de espacios de alta dimensionalidad sobre unidades de mapa de tal manera que esa distancia relativa entre los datos puntuales es preservada.

Las unidades del mapa, o neuronas, forman usualmente una cuadrícula regular de dos dimensiones. Cada neurona del SOM está representada por un peso de dimensión n, o vector modelo $\mathbf{v}_i = [v_{i,n}, \ldots, v_{i,n}]^T$. Estos vectores de peso del SOM forman un código cifrado. Las neuronas del mapa están conectadas a neuronas adyacentes por una relación de vecindad, lo cual dicta la topología del mapa. El número de neuronas determina el granulado del mapeo, lo cual afecta la exactitud y la capacidad de generalización del SOM.

SOM es un vector cuantificador, dónde los pesos juegan el papel de los vectores de código cifrado. Esto quiere decir, cada vector de peso representa un vecindario local del espacio, también llamado celda Voronoi. La respuesta de un SOM a una entrada x es determinada por el vector referencia (peso) $\mathbf{v}_i^0$ que produzca la mejor equivalencia de la entrada

$$i^0 = \arg_i \min ||\mathbf{v}_i - x|| \tag{1.55}$$

Donde i^0 representa el índice de la Mejor Unidad Pareada (BMU por sus siglas en ingles). Durante el entrenamiento iterativo, el SOM forma una red elástica que hace pliegues encima de la nube formada por los datos. La red tiende a aproximar la densidad de probabilidad de los datos: Los vectores de código cifrado tienden al trasfondo allí donde los datos son densos, mientras donde solamente haya unos

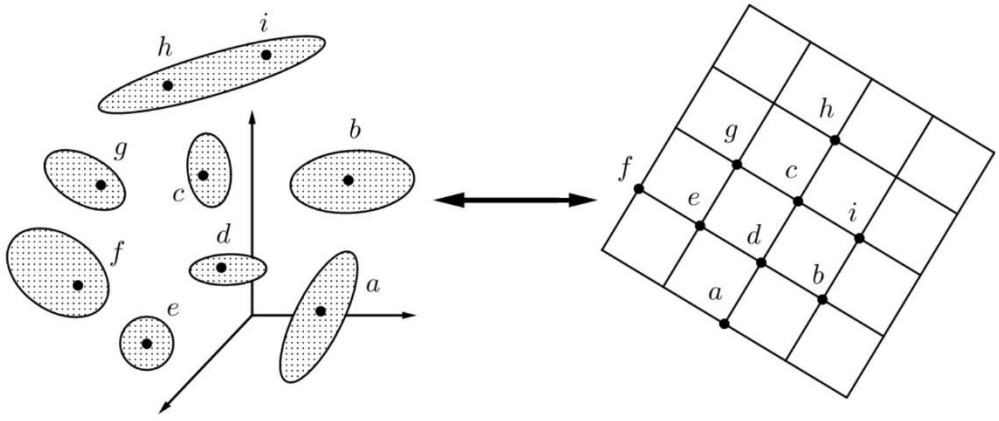


Figura 1.9: Ilustración de la conservación de la topología propiedad de SOM.

pocos vectores de código cifrado donde los datos son escasos.
El entrenamiento de SOM puede estar completo generalmente con una regla competitiva de aprendizaje como

$$\mathbf{v}_i^{(k+1)} = \mathbf{v}_i^{(k)} + \eta \Lambda_{i^0,i}(\mathbf{x} - \mathbf{v}_i^{(k)}) \tag{1.56}$$

Donde $\Lambda_{i^0,i}$ es una función de vecindario espacial y η es la tasa de aprendizaje. Por lo general, la función del vecindario es

$$\Lambda_{i^0,i} = \exp\left(\frac{||\mathbf{r}_i - \mathbf{r}_i^0||^2}{2\sigma^{2(k)}}\right) \tag{1.57}$$

Donde $||\mathbf{r}_i - \mathbf{r}_i^0||$ representa la distancia Euclidiana en el espacio de salida entre el vector del $i-$ésimo y el ganador.
Todo el procedimiento quiere ilustrarse en la Figura 1.9 y la Figura 1.10.

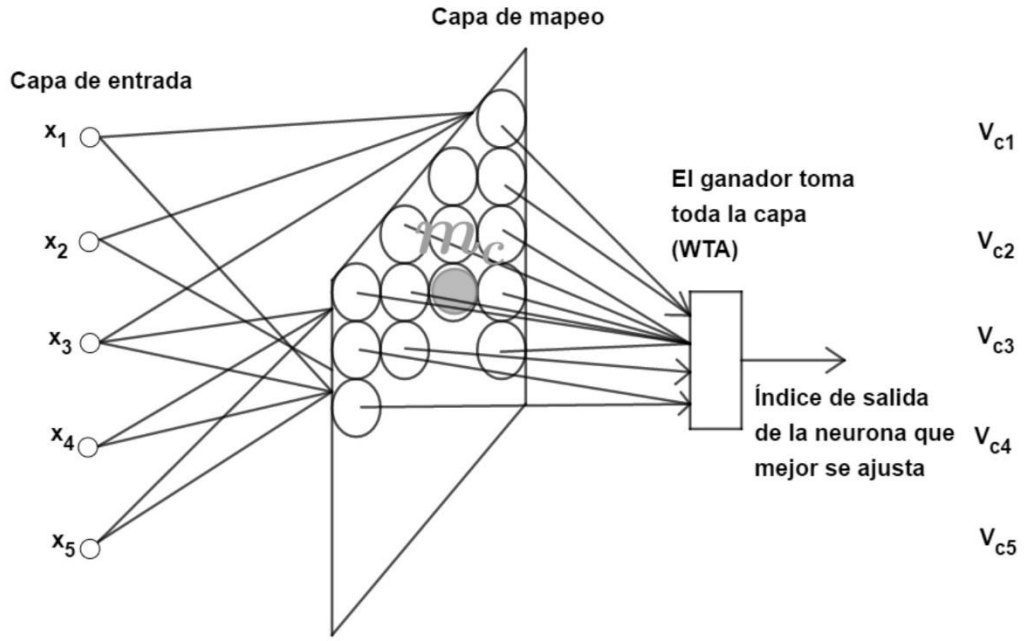


Figura 1.10: Ilustración de un BMU computacional.

Usualmente, con el uso de esta herramienta los centros de los grupos (el código cifrado de la SOM) son mapeados en un espacio de dos dimensiones.
Recientemente, varios avances descritos han sido propuestos para aumentar el desempeño de SOM por la incorporación de lógica difusa. Las neuronas son reemplazadas por reglas difusas que permiten un modelaje eficiente de funciones de valores continuos.
En algoritmos de cuantización para vectores, el agrupamiento difuso combinado con SOM se usa para proyectar los datos a dimensiones bajas. En trabajos de agrupamiento difuso de Kohonen, algunos aspectos del modelo difuso c-medias son integrados en el tipo Kohonen agrupamiento clásico de armazón dura.
Otro acercamiento se ha replanteado en medios difusos suavemente distribuídos, dónde un mapa auto-estructurado difuso es desarrollado basado en las modificaciones funcionales difusas de c-medias. El análisis de agrupamientos difusos

de c-medias también ha estado combinado con mapeos similares y exitosamente aplicado al mapa de la distribución de contaminantes y a rastrear sus fuentes para tener acceso a riesgos potenciales ambientales en una base de datos del suelo de Austria.

Ejemplo 4 ***(La Visualización de datos de Vino basados en Mapas Auto-organizados).*** *El SOM ha sido utilizado para visualizar los datos de Vino. SOM puede eficazmente servir para correlación cruzada, dicho procedimiento es útil para detectar las características redundantes. Es interesante para notar que esas reglas pueden ser dadas por el mapa de las variables dado en la Figura 1.11.*
Por ejemplo, si Alcohol es alto y Flavonoides está alto y la intensidad de Color es media, entonces el vino es de clase 1. Eso es visible también en los datos. Este conocimiento puede ser fácilmente validado analizando el SOM de los datos dados en la Figura 1.11.

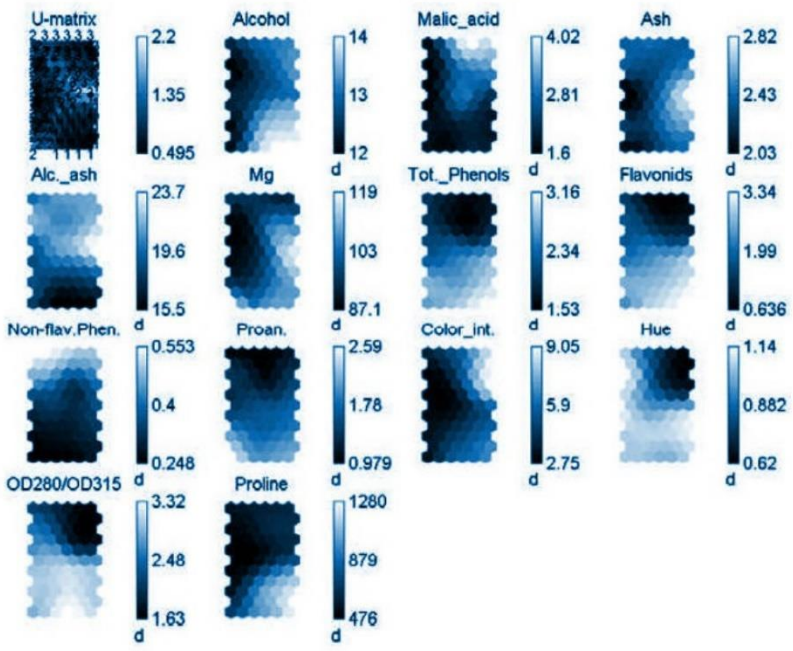


Figura 1.11: Mapas auto-organizados de los datos de Vino.

Existen otros muchos métodos aparte de los pocos descritos arriba.

- **La proyección Persecución.** La proyección Persecución (PP) es una técnica no supervisada que se interesa en investigar proyecciones lineales de baja

dimensión en datos de dimensional alto optimizando una cierta función objetivo llamada Índice de Proyección (PI por sus siglas en ingles). El índice de proyección define el intento del procedimiento PP. La notación de interés obviamente varía con la aplicación. La meta de la minería de datos (es decir, revelando una tendencia del agrupamiento de datos, un borde o salto de densidad de los datos) debería ser traducido a un índice numérico, siendo una proyección funcional de la distribución de los datos.

Esta función debería alterarse continuamente con los parámetros definiendo la proyección y tener un valor grande cuando la distribución proyectada está definida para ser interesante y pequeña de otro caso.

La mayoría de índices de proyección son desarrollados desde el punto de vista que la normalidad representa la noción de "poco interesante".

Difieren en las suposiciones acerca de la naturaleza de la desviación de la normalidad y en su eficiencia computacional. Generalmente, pueden estar divididos en dos grupos: Los índices paramétricos de proyección y los no paramétricos. Los índices paramétricos de proyección son diseñados para captar cualquier salida de distribución de datos de una distribución especificada, mientras que los índices no paramétricos son más generales y ellos no están enfocados a una distribución particular.

- **Los Mapas Topográficos Generativos.** El Mapeo Topográfico generativo (GTM por sus siglas en ingles), presentado por Bishop Et Al., puede ser considerado como una reformulación probabilística del acercamiento del mapa autoestructurador (SOM). La meta del procedimiento GTM es modelar la distribución de datos en un espacio dimensional alto en términos de un número más pequeño de variables latentes.

- **Redes de avance auto-asociativo.** Hacia adelante es usualmente usada en trasfondos supervisados. No obstante, puede ser aplicado como un método no lineal de proyección. En este caso la red está adiestrada para trazar un

mapa de un vector de sí mismo a través de un estrato cuello de botella, es decir, el estrato con un número más pequeño de nodos que el estrato de entrada (y de salida). El estrato del cuello de botella de la red realiza la reducción de la dimensión, porque el número neuronas en este estrato es más pequeño que en la entrada y el estrato de salida, a fin de que la red se ve forzada a desarrollar una representación compacta de los datos de entrada. La (figura 1.12) muestra un cuadro esquemático de una red auto-asociativa.

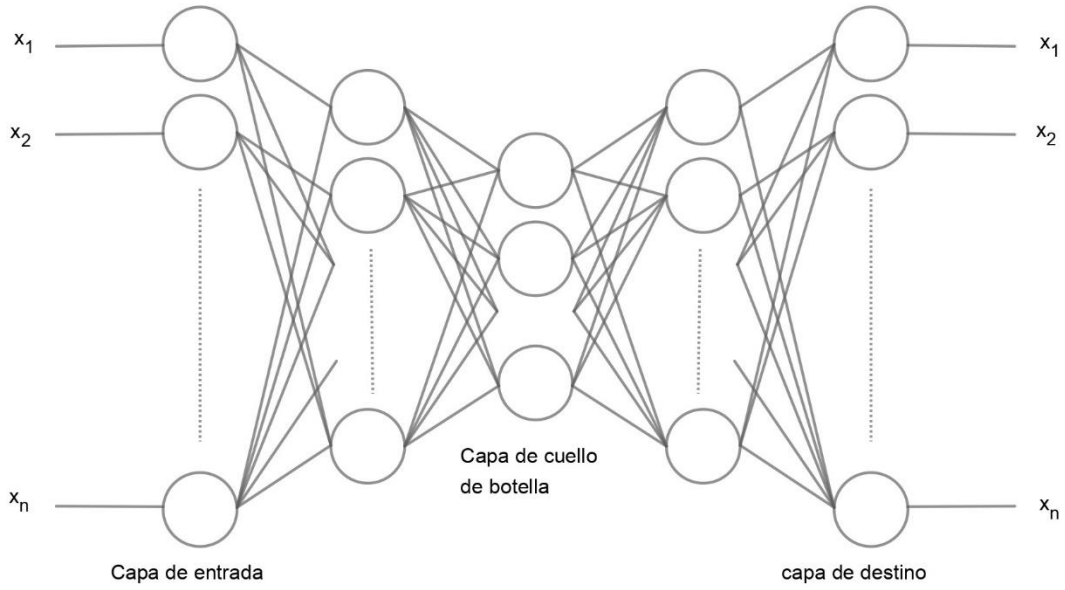


Figura 1.12: Red PCA no Lineal Autoasociativa.

Si todas las unidades en esta red son ocupadas para ser lineales, en cuyo caso cualquier capa intermedia entre las entradas y los objetivos y el estrato del cuello de botella pueden ser removidos, y la red estará adiestrada usando la función suma de los errores al cuadrado, este entrenamiento es propio de la minimización del error de reconstitución en (2.4). Esto resultará en el sistema de redes realizando un ACP estándar. De hecho, puede ser mostrado también en el caso para una red con un solo estrato de cuello de botella de

unidades no lineales.

- **Análisis discriminante.** La proyección del análisis discriminante maximiza la dispersión entre los grupos mientras se mantiene la dispersión dentro del grupo. Esta proyección requiere que todos los patrones tengan etiquetas de clase o etiquetas de patrón. Incluso cuando no hay etiquetas de categoría extrínsecas disponibles, los patrones se pueden agrupar y las etiquetas de grupo se pueden usar como información de categoría para fines de proyección. Ver la siguiente figura.

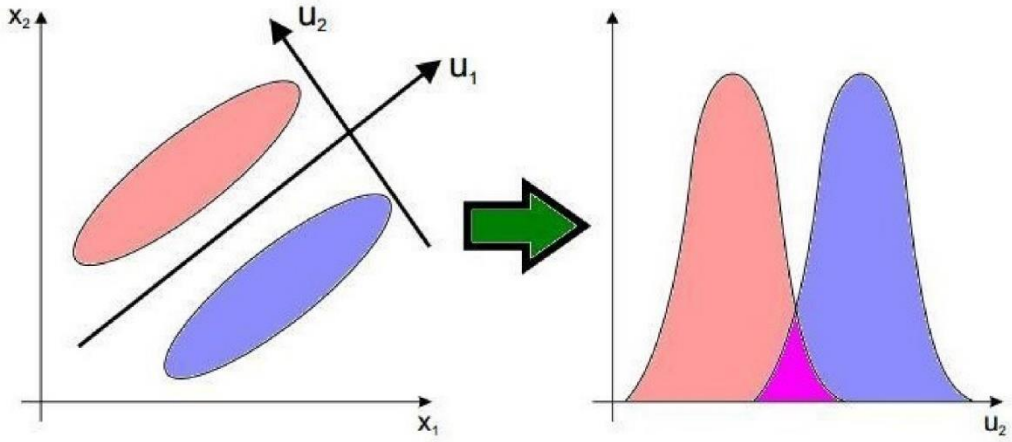


Figura 1.13: Esquema del análisis discriminante.

El escalamiento multidimensional es un nombre genérico para un conjunto de procedimientos y algoritmos que comienzan con una matriz de proximidad ordinal y generan configuraciones de puntos en una, dos o tres dimensiones.

La escala multidimensional traduce una escala ordinal a un conjunto de escalas de relación y es un ejemplo de ordenación. MDSCAL desarrollado por Kruskal, es una de las técnicas más populares en este campo. Dado que el objetivo de un método de escalamiento multidimensional es crear un conjunto de escalas o dimensiones que representan los datos, existen vínculos

naturales entre el escalamiento multidimensional, la dimensionalidad intrínseca y la proyección no lineal.

Las técnicas mencionadas hasta ahora en este capítulo son métodos de visualización general, que no tienen una conexión directa con el agrupamiento (excepto SOM). En las siguientes dos secciones se presentarán nuevos métodos para la visualización de los resultados de la agrupación. El primer método se basa en los resultados de los algoritmos clásicos de agrupación difusa y se aplica un método de proyección iterativo para preservar la estructura de datos en dos dimensiones en el sentido que las distancias entre los puntos de datos y los centros de agrupación deben ser similares en el espacio original y en las dos dimensiones proyectadas también. El segundo enfoque es un método de agrupamiento difuso modificado y el propósito de la modificación es aumentar la aplicabilidad del algoritmo clásico c-medias difuso ordenando los centros de agrupamiento (prototipos) en un espacio de baja dimensionalidad fácilmente visualizable.

1.2. Visualización de resultados de agrupamiento difuso por Mapeo de Sammon modificado.

Esta sección se centra en la aplicación del mapeo de Sammon para la visualización de los resultados del agrupamiento, ya que el mapeo de las distancias está mucho más cerca de la tarea del agrupamiento que de preservar las variaciones. Hay dos problemas principales encontrados en la aplicación del mapeo de Sammon a la visualización de resultados del agrupamiento difuso:

- Los prototipos de algoritmos de agrupamiento pueden ser vectores (centros) de la misma dimensión que los objetos de datos, pero también pueden definirse como objetos geométricos de "nivel superior", como sub-espacios lineales o no lineales o funciones. Por lo tanto, los métodos de proyección clásicos basados en la varianza de los datos (ACP) o basados en la preservación de

la distancia euclidiana entre los puntos de los datos (mapeo de Sammon) no son aplicables cuando los algoritmos de agrupamiento no utilizan la norma de la distancia euclidiana.

- Como el mapeo de Sammon intenta preservar la estructura de los n datos de alta dimensionalidad al encontrar N puntos en un espacio de q datos de baja dimensionalidad, donde las distancias entre los puntos medidos en el espacio q-dimensional se aproximan a las distancias correspondientes entre los puntos en el espacio n-dimensional, el algoritmo implica una gran cantidad de cálculos, ya que en cada iteración requiere el cálculo de $N(N-1)/2$ distancias. Por lo tanto, la aplicación del mapeo de Sammon se vuelve impráctico para N grande.

Al utilizar las propiedades básicas de los algoritmos de agrupamiento difuso, una idea útil y fácilmente aplicable es mapear los centros de agrupación y los datos de modo que se mantengan las distancias entre las agrupaciones y los puntos de datos (ver Figura 1.14). Durante el proceso de mapeo iterativo, el algoritmo usa los valores de pertenencia de los datos y minimiza una función objetivo que es similar a la función objetivo del algoritmo de agrupamiento original.

1.2.1. Mapeo de Sammon Modificado.

Para evitar el problema mencionado anteriormente, a continuación presentamos algunas modificaciones para adaptar el mapeo de Sammon para la visualización de resultados del agrupamiento difuso. Mediante el uso de las propiedades básicas de los algoritmos de agrupamiento difuso donde la distancia entre los puntos de datos y los centros de agrupación se consideran importantes, el algoritmo modificado toma en cuenta solamente las distancias $N \times c$, donde c representa

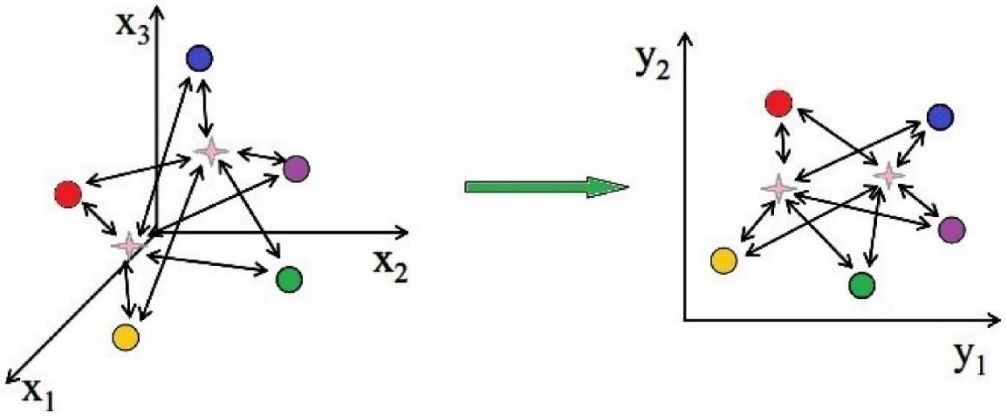


Figura 1.14: Ilustración del método de Sammon difuso.

el número de clúster, ponderados por los valores de pertenencia:

$$E_{fuzz} = \sum_{i=1}^{c} \sum_{k=1}^{N} (\mu_{i,k})^m (d(\mathbf{x}_k, \eta_i) - d^*(\mathbf{y}_k, \mathbf{z}_i))^2. \tag{1.58}$$

Donde $d(\mathbf{x}_k, \eta_i)$ representa la distancia entre los $\mathbf{x}_k$ datos puntuales y los η_i centros de los clústers medidos en el espacio original n-dimensional, mientras $d^*(\mathbf{y}_k, \mathbf{z}_i)$ representa la distancia euclidiana entre el centro del clúster proyectado $\mathbf{z}_i$ y los datos proyectados $\mathbf{y}_k$. Esto significa que, en el espacio proyectado, cada agrupación está representada por un solo punto, independientemente de la forma del prototipo de agrupación original, η. La aplicación de la medida de distancia Euclidiana simple aumenta la interpretabilidad de los resultados gráficos (normalmente en dos dimensiones, aunque también se pueden utilizar gráficos tridimensionales). Si el tipo de prototipos de grupos se selecciona correctamente, los datos proyectados caerán cerca del centro del grupo proyectado representado por un punto que resulta en un grupo de forma aproximadamente esférica (ver Algoritmo 2.2.1).

El algoritmo resultante es similar al mapeo original de Sammon, pero en este caso, en cada iteración posterior a la adaptación de los puntos de datos proyectados, los centros de clústers proyectados se recalculan en función de la fórmula de la media ponderada de los algoritmos de agrupamiento difuso.

El gráfico bidimensional resultante de los datos proyectados y los centros de agru-

pamiento son fácilmente interpretables, ya que se basan en las medidas de distancia euclidianas normales entre los centros de agrupamiento y los puntos de datos. Sobre la base de estas distancias mapeadas, los valores de pertenencia de los datos proyectados también se pueden trazar en función de la fórmula clásica del cálculo de los valores de pertenencia:

$$\mu_{i,k}^{*} = \frac{1}{\sum_{i=1}^{c} \left(\frac{d*(\mathbf{x}_k, \eta_i)}{d*(\mathbf{x}_k, \eta_j)} \right)^{\frac{2}{m-1}}} \tag{1.59}$$

Por supuesto, el gráfico resultante solo se aproximará al problema de agrupamiento de alta dimensionalidad original. La calidad de la aproximación puede evaluarse fácilmente basándose en el error cuadrático medio del original y los valores de pertenencia recalculados.

$$\mathbf{P} = ||\mathbf{U} - \mathbf{U}^{*}|| \tag{1.60}$$

donde $\mathbf{U} = \mu_{i,k}^{*}$ representa la matriz de pertenencia recalculada. Por supuesto, hay otras herramientas para obtener información sobre la calidad del mapeo de los clúster. Por ejemplo, la comparación de las medidas de validación de clúster calculadas en función de los valores originales y el mapeo de los valores de pertenencia también se puede usar para este propósito.

Capítulo 2

Agrupamiento para la Identificación de Modelos Difusos

2.1. Introducción al Modelamiento Difuso

Para muchas aplicaciones del mundo real, una gran cantidad de información es proporcionada por humanos expertos, quienes no razonan en términos matemáticos, pero en su lugar describen el sistema verbalmente mediante declaraciones vagas o comparaciones imprecisas.

Si *la temperatura es elevada,* ***entonces*** *la presión es alta* (2.1)

Esto es porque tanta experiencia y conocimiento humano son dados en términos de reglas verbales, uno de los avances atinados de la ingeniería es tratar de integrar tal información lingüística en el proceso modelador. Un acercamiento muy común y conveniente de hacer esto consiste en usar conceptos de lógica difusa para repartir el conocimiento verbal en una representación matemática convencional (estructura del modelo), lo cual subsiguientemente puede ser puesto a punto usando datos de entrada y salida.

La lógica difusa propuesta por primera vez por Lotfi Zadeh en 1965 [7] ante todo concierne en la representación del conocimiento algo impreciso que es común en los sistemas naturales. Facilita la representación en computadoras digitales de este tipo de conocimiento a través del uso de conjuntos difusos. Desde esta base,

la lógica difusa emplea los operadores lógicos para cotejar e integrar este conocimiento y aproximar el tipo de razonamiento común en la inteligencia natural.

Un modelo difuso consiste en una armazón computacional basada en los conceptos de conjuntos difusos, reglas difusas si - entonces, y el razonamiento difuso. No pretenderá proveer un reconocimiento amplio del campo. Para tal sondeo el lector es referido a "An Introduction to Fuzzy Control" de Driankov, Hellendoorn, y Reinfrank [10] o "Fuzzy Control" de K.M. Passino y S. Yurkovic [11], o "A course in Fuzzy Systems and Control" de L.X. Wang[9].
Convencionalmente la teoría de conjuntos se basa sobre la premisa de que un elemento pertenece o no está incluido en un conjunto dado. La teoría de los conjuntos difusos toma una visión menos rígida y da permiso de colocar grados de pertenencia a los elementos que son algo semejante entre si y no están restringidos para estar o no en un conjunto, pero tienen permiso de estar "algo" adentro. En muchos casos, este es un enfoque más natural. Por ejemplo, considere el caso de una persona que describe la temperatura atmosférica como "caliente". Si uno expresara este concepto en la teoría convencional de conjuntos, se vería obligado a designar un rango distinto de temperaturas, como $25°C$ o más, como pertenecientes al conjunto caliente. Es decir:

$$caliente = [25°C, \infty)$$

Esto parece inventado porque cualquier temperatura que cae ligeramente fuera de este rango no sería un miembro del conjunto, a pesar de que un ser humano puede no ser capaz de distinguir entre uno y el que está justo dentro del conjunto. En la teoría de los conjuntos difusos, no se impone una representación precisa del conocimiento impreciso ya que no se requiere que se definan los límites estrictos de un conjunto, sino que se define una función de pertenencia. Una función de pertenencia describe la relación entre una variable y el grado de pertenencia al conjunto difuso, eso corresponde a los valores particulares de esa variable. Este

grado de pertenencia es usualmente definido en términos de un número entre 0 y 1, inclusive, donde 0 implica ausencia total de pertenencia, 1 implica pertenencia completa, y cualquier valor intermedio implica pertenencia parcial del conjunto difuso. Esto puede ser escrito como sigue:

$$A(x) \in [0,1] \ para \ x \in U$$

Donde $A(\cdot)$ es la función de pertenencia y U es el universo de discurso que define el rango total de interés sobre el cual la variable x debería estar definida.
Por ejemplo, para definir la pertenencia del conjunto difuso "caliente", una función que asciende de 0 para 1 sobre el rango $15°C$ a $25°C$ puede ser usada, es decir,

$$A(x) = \begin{cases} 0, & si \ x < 15°C \\ \frac{x-15}{10}, & si \ 15°C \leq x \leq 25°C \\ 1, & si \ x > 25°C \end{cases}$$

Esto implica que $15°C$ no está caliente; $20°C$ está un poco caliente; $23°C$ está muy caliente; Y $30°C$ está verdaderamente caliente. Los valores medibles específicos, como 15 y 20 son a menudo llamados valores puntuales o difusos únicos, para distinguirlos de valores difusos, tan calientes que son definidos por un conjunto difuso. Los valores difusos son algunas veces también llamados valores lingüísticos.

Como ilustra la Figura 2.1, esta definición refleja más las interpretaciones humanas o lingüísticas de las temperaturas y, por lo tanto, se aproxima mejor a dichos conceptos.

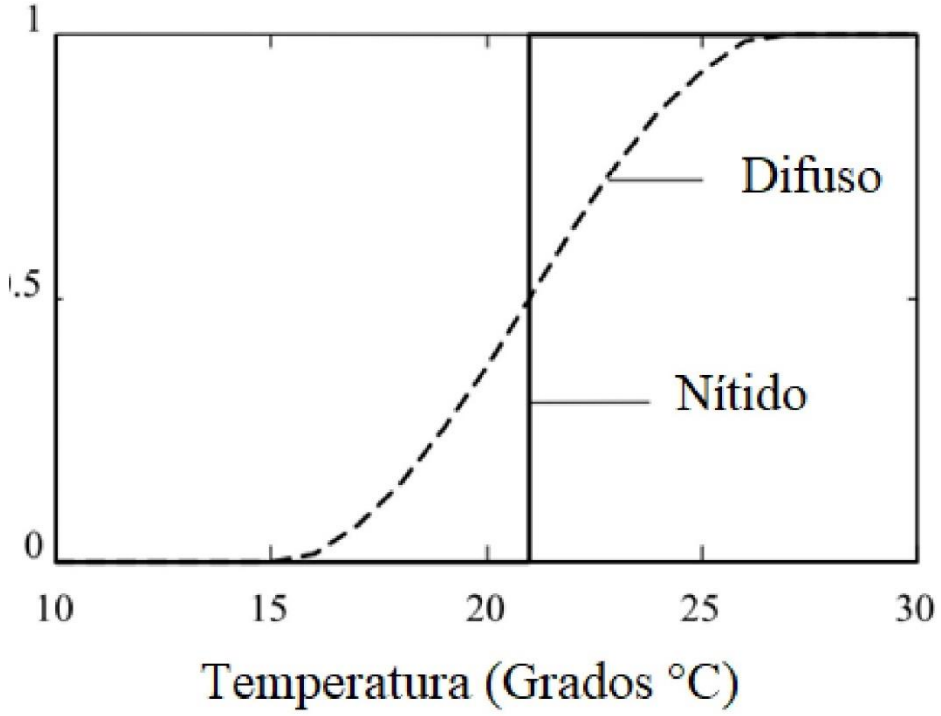


Figura 2.1: Representación de las temperaturas altas.

Si bien parece impreciso para un ser humano, los conjuntos difusos son matemáticamente precisos en que pueden ser totalmente representados por números exactos. Por consiguiente, pueden verse como un método de vincular al humano y las representaciones de conocimiento de la máquina.

Dado como un método natural de representar información en una computadora existente, los métodos de procesamiento de información pueden ser aplicados por y para el uso de modelos difusos.

La configuración básica de un modelo difuso se muestra en la figura 2.2,

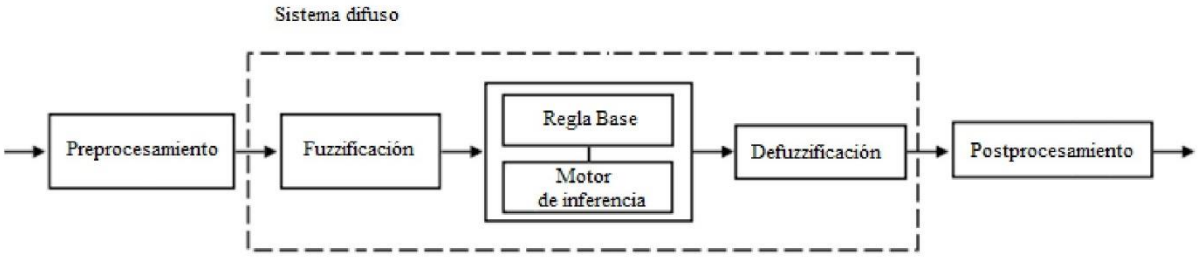


Figura 2.2: Estructura de un sistema difuso.

En esta figura, el modelo difuso involucra los siguientes componentes:

- **Preprocesamiento de datos**. Los valores físicos de la entrada del sistema difuso pueden diferir significativamente en magnitud. Al asignarlos a dominios normalizados (pero interpretables) adecuados a través del escalado, uno puede trabajar con señales aproximadamente de la misma magnitud, lo cual es deseable desde el punto de vista de la estimación.

- **Fuzzificación**. La fuzzificación asigna los valores nítidos de la entrada preprocesada del modelo en conjuntos difusos adecuados representados por funciones de pertenencia (MF).

  A medida que el antecedente y consecuente de los conjuntos difusos toman significados lingüísticos como "alta temperatura", se denominan etiquetas lingüísticas de los conjuntos de variables lingüísticas. Por ejemplo, si la variable lingüística es "temperatura", se pueden definir varios conjuntos difusos para esta variable, por ejemplo, "bajo", "medio", "alto", etc. ver Figura 2.1.

  El grado de pertenencia de una sola variable nítida a un solo conjunto difuso podría evaluarse mediante una función de pertenencia. Un difusor calcula el grado de pertenencia de múltiples variables nítidas a múltiples conjuntos difusos en la forma de uno a muchos. Hay $n \geq 1$ variables de entrada nítidas y cada variable puede pertenecer a $M_i > 1 : i = 1, \ldots, n$ conjuntos difusos.

  Por ejemplo, un sistema de aire acondicionado puede tener dos variables de

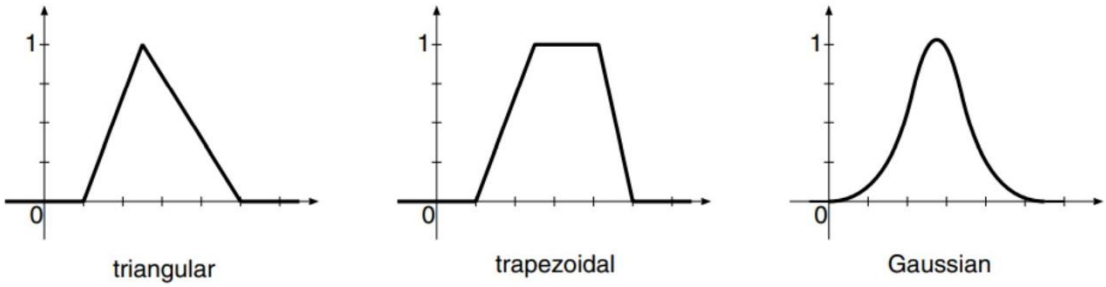


Figura 2.3: Tres formas comunes de funciones de pertenencia.

entrada nítidas, temperatura y humedad, es decir,$n = 2$. Estos pueden transformarse en dos variables difusas que consisten en los conjuntos difusos frio, fresco, tibio, cálido, caliente y seco, normal, caliente, respectivamente. Esto significa que $M_1 = 5$ y $M_2 = 3$.
Los sistemas de una sola variable de entrada son factibles, pero es bastante evidente que, si solo se define un conjunto difuso para una variable de entrada en particular, entonces no se pueden hacer distinciones en las reglas de esta variable y su inclusión en el modelo difuso es redundante. Por lo tanto, generalmente se definirán dos o más conjuntos difusos para cada variable de entrada.

Ya se ha mencionado que el grado de pertenencia de una variable nítida a un conjunto difuso se define mediante una función de pertenencia. En este trabajo se utilizarán exclusivamente las funciones de pertenencia triangulares. La función de pertenencia triangular y algunas otras formas de función de pertenencia comúnmente utilizadas se muestran en la Figura 2.3.

- **Regla Base**. La regla base es la piedra angular del modelo difuso. El conocimiento experto, que se supone que se proporciona como una serie de reglas **Si-Entonces**, se almacenan en una base de reglas difusas.
  En los sistemas difusos basados en reglas, la relación entre las variables se

representa mediante las reglas **Si-Entonces** de la siguiente forma general:

$$\textit{Si}\ \textit{la proposición antecedente}\ \textit{entonces}\ \textit{la proposición consecuente} \tag{2.2}$$

Según la proposición consiguiente y la estructura de las reglas bases, hay tres clases distintas de modelos difusos:

1. **Modelos lingüísticos difusos** (modelos Mandani) donde tanto el antecedente como el consecuente son proposiciones difusas. Por lo tanto, una regla general de un modelo difuso lingüístico o Mandani viene dado por:

$$R_j : \textbf{Si}\ x_1 \text{ es } A_{1,j} \text{ y } \ldots \text{ y } x_n \text{ es } A_{n,j}\ \textbf{entonces}\ y \text{ es } B_j \tag{2.3}$$

   Donde R_j denota la j-ésima regla, $j = 1, \ldots, N_r$, y N_r es el número de reglas. Las variables antecedentes representan la entrada de los sistemas difusos $\mathbf{x}$. $A_{i,j}$ y B_j son conjuntos difusos descritos por funciones de pertenencia $\mu_{A_{i,j}}(x) :\rightarrow [0,1]$ y $\mu_{B_i}(y) :\rightarrow [0,1]$.

2. **Modelos relacionales difusos** se basan en relaciones difusas y ecuaciones relacionales. Estos modelos pueden ser considerados como una generalización del modelo lingüístico, permitiendo que una proposición antecedente particular se asocie con varias proposiciones consecuentes diferentes a través de una relación difusa.

3. **Takagi–Sugeno (TS)** modelos difusos donde el consecuente es una función crisp de las variables de entrada, $f_j(x)$, en lugar de una proposición difusa

$$R_j : \textbf{Si}\ x_1 \text{ es } A_{1,j} \text{ y } \ldots \text{ y } x_n \text{ es } A_{n,j}\ \textbf{entonces}\ y \text{ es } f_j(\mathbf{x}) \tag{2.4}$$

   Esta tesis trata de este tipo de modelos difusos.

- **Máquina de inferencia**. El mecanismo de inferencia o motor de inferencia es el método computacional que calcula el grado en que cada regla se dispara

para un patrón de entrada difuso dado considerando los conjuntos de reglas y etiquetas. Se dice que una regla se activa cuando se dan las condiciones de las que depende. Dado que estas condiciones están definidas por conjuntos difusos que tienen grados de pertenencia, una regla tendrá un grado de activación o fuerza de activación, β_j. La fuerza de disparo está determinada por el mecanismo que se utiliza para implementar el **y** en la expresión (2.4); se utilizará el producto de los grados de pertenencia, es decir:

$$\beta_j = \prod_{i=1}^{n} A_{i,j} \tag{2.5}$$

Donde $A_{i,j}$, define la función de pertenencia en la entrada i se usa en la regla j. Nuevamente, existen diferentes métodos para implementar cada uno de los operadores lógicos.

- **Defuzzificación**. Un defuzzificador compila la información proporcionada por cada una de las reglas y toma una decisión a partir de esta base. En los modelos difusos lingüísticos, la difusificación convierte los conjuntos difusos resultantes definidos por el motor de inferencia a la salida del modelo en una señal crisp estándar. El método que se utiliza en este trabajo es el método comúnmente llamado método del centro de gravedad o centroide. En el caso de modelos difusos TS, se describe mediante la siguiente ecuación:

$$y = \frac{\sum_{j=1}^{N_r} \beta_j f_j(\mathbf{x})}{\sum_{j=1}^{N_r} \beta_j}. \tag{2.6}$$

Puede verse que el método centroide de difusificación toma una suma ponderada de las consecuencias designadas de las reglas de acuerdo con las fuerzas de activación de las reglas. Hay muchos otros tipos de difusificadores como el centro de las sumas, el primero de los máximos y el medio de los máximos.

- **Postprocesamiento**. El paso de postprocesamiento proporciona la salida del sistema difuso en función de la señal crisp obtenida después de la defuzzificación. Esto a menudo significa el escalamiento de la salida.

Se definen las funciones de base difusa como:

$$\beta_j(\mathbf{x}) = \frac{\prod_{i=1}^{n} A_{i,j}(x_i)}{\sum_{j=1}^{N_r} \prod_{i=1}^{n} A_{i,j}(x_i)} \tag{2.7}$$

$$y = \sum_{j=1}^{N_r} \beta_j(\mathbf{x})\theta_j \tag{2.8}$$

Donde $\beta_j(\mathbf{x})$ es la fuerza de disparo normalizada de la regla j, $A_{i,j}(x_j)$ representa una función de pertenencia, x_i es la entrada i-ésima y θ_j es la regla precisa consecuente, $f_j(x) = \theta_j$.

Sea Y el conjunto de todas las expansiones (2.8) con $\beta_j(\mathbf{x})$ dado por (2.7) y $d_\infty(f_1, f_2) = sup_{x \in U}|f_1(x) - f_2(x)| < \epsilon$ la métrica del supremo, entonces (Y, d_∞) es un espacio métrico[8].

Wang y Mendel prueban que, con suficientes reglas, este sistema puede aproximar cualquier función continua real a cualquier precisión dada; esto se afirma de la siguiente manera:

Teorema 2 *Dada cualquier función continua $f(\cdot)$ en el conjunto compacto $U \subset \mathbb{R}^n$ y una constante arbitraria $\epsilon > 0$, existe una función $\hat{f}(\cdot)$, definida en el conjunto de todas las expansiones de funciones básicas difusas, de modo que:*

$$min(x_1, \ldots, x_n) \in U|\hat{f}(x_1, \ldots, x_n) - f(x_1, \ldots, x_n)| < \epsilon$$

Donde $\hat{f}(x_1, \ldots, x_n)$ es la función implementada por la función de bases difusa.

Usamos el siguiente teorema de Stone-Weierstrass para probar el teorema.

Teorema de Stone-Weierstrass: Sea Z un conjunto de funciones continuas reales en un conjunto compacto U. Si

a) Z es un álgebra, es decir, el conjunto Z se cierra bajo suma, multiplicación y multiplicación escalar,

b) Z separa puntos en U, es decir, por cada $x, y \in U, x \neq y$, existe $f \in Z$ tal que $f(x) \neq f(y)$, y

c) Z desaparece en ningún punto de U, es decir, por cada $x \in U$ existe $f \in Z$ tal que $f(x) \neq 0$, entonces el cierre uniforme de Z consiste en todas las funciones continuas reales en U; es decir, (Z, d_∞) es denso en $(C[U], d_\infty)$.

Prueba: Primero, probamos que (Y, d_∞) es un álgebra. Deje $f_1, f_2 \in Y$, para que podamos escribirlos como

$$f_1(x) = \frac{\sum_{j=1}^{K1} \left(\bar{z}1^j \prod_{i=1}^{n} \mu_{A1_i^j}(x_i) \right)}{\sum_{j=1}^{K1} \left(\prod_{i=1}^{n} \mu_{A1_i^j}(x_i) \right)} \tag{2.9}$$

$$f_2(x) = \frac{\sum_{j=1}^{K_2} \left(\bar{z}2^j \prod_{i=1}^{n} \mu_{A2_i^j}(x_i) \right)}{\sum_{j=1}^{K2} \left(\prod_{i=1}^{n} \mu_{A2_i^j}(x_i) \right)} \tag{2.10}$$

por lo tanto, tenemos

$$f_1(x) + f_2(x) = \frac{\sum_{j1=1}^{K1} \sum_{j2=1}^{K2} \left(\bar{z}1^{j1} + \bar{z}2^{j2} \right) \left(\prod_{i=1}^{n} \mu_{A1_i^{j1}}(x_i)\, \mu_{A2_i^{j2}}(x_i) \right)}{\sum_{j1=1}^{K1} \sum_{j2=1}^{K2} \left(\prod_{i=1}^{n} \mu_{A1_i^{j1}}(x_i)\, \mu_{A2_i^{j2}}(x_i) \right)} \tag{2.11}$$

Desde $\mu_{A1_{ij1}}$ y $\mu_{A2_{ij2}}$ son de forma gaussiana, su producto $\mu_{A1_{ij1}} \mu_{A2_i^{j2}}$ también tiene forma gaussiana (esto se puede verificar mediante operaciones algebraicas sencillas); por tanto, (2.11) tiene la misma forma que

$$f_1(x) f_2(x) = \frac{\sum_{j1=1}^{K1} \sum_{j2=1}^{K2} \left(\bar{z}1^{j1} \bar{z}2^{j2} \right) \left(\prod_{i=1}^{n} \mu_{A1_i^{j1}}(x_i)\, \mu_{A2_i^{j_i}}(x_i) \right)}{\sum_{j1=1}^{K1} \sum_{j2=1}^{K2} \left(\prod_{i=1}^{n} \mu_{A1_i^{j1}}(x_i)\, \mu_{A2_i^{j_i}}(x_i) \right)}, \tag{2.12}$$

de modo que $f_1 + f_2 \in Y$. De manera similar, tenemos (2.13), que también tiene la misma forma que (2.8); por tanto, $f_1 f_2 \in Y$. Finalmente, para $c \in R$,

$$cf_1(x) = \frac{\sum_{j=1}^{K1} \left(c\bar{z}1^j\right) \left(\prod_{i=1}^{n} \mu_{A1_i^j}(x_i)\right)}{\sum_{j=1}^{K1} \left(\prod_{i=1}^{n} \mu_{A1_i^j}(x_i)\right)} \tag{2.13}$$

que de nuevo tiene la forma de (2.8); por tanto, $cf_1 \in Y$. Por tanto, (Y, d_∞) es un álgebra.

A continuación, probamos que (Y, d_∞) separa puntos en U. Demostramos esto construyendo un f requerido; es decir, especificamos $f \in Y$ tal que $f(x^0) \neq f(y^0)$ para $x^0, y^0 \in U$ dados arbitrariamente con $z^0 \neq y^0$. Elegimos dos reglas difusas en la forma de (2.10) para la base de reglas difusas (es decir, $M = 2$). Sea $x^0 = (x_1^0, x_2^0, \cdots, x_n^0)$ y $y^0 = (y_1^0, y_2^0, \cdots, y_n^0)$. If $x_i^0 \neq y_i^0$, definimos dos conjuntos difusos $\left(A_i^1, \mu_{A_i^1}\right)$ y $\left(A_i^2, \mu_{A_i^2}\right)$ con

$$\mu_{A_i^1}(x_i) = \exp\left[-\frac{\left(x_i - x_i^0\right)^2}{2}\right] \tag{2.14}$$

$$\mu_{A_i^2}(x_i) = \exp\left[-\frac{\left(x_i - y_i^0\right)^2}{2}\right]. \tag{2.15}$$

Si $x_i^0 = y_i^0$, entonces $A_i^1 = A_i^2$ y $\mu_{A_i^1} = \mu_{A_i^2}$; es decir, solo se define un conjunto difuso. Definimos dos conjuntos difusos $\left(B^1, \mu_{B^1}\right)$ y $\left(B^2, \mu_{B^2}\right)$ con

$$\mu_{B^j}(z) = \exp\left[-\frac{\left(z - \bar{z}^j\right)^2}{2}\right], \tag{2.16}$$

donde $j = 1, 2$, y $\bar{z}^j$ se especificarán más adelante. Ahora hemos especificado todos los parámetros de diseño excepto $\bar{z}^j (j = 1, 2)$, es decir, ya hemos obtenido una función f que tiene la forma de (2.8) con $M = 2$ y $\mu_{A_i^j}$ dado por (2.14) y (2.15).

Con este f, tenemos

$$f(x^0) = \frac{\bar{z}^1 + \bar{z}^2 \prod_{i=1}^{n} \exp\left[-\left(x_i^0 - y_i^0\right)^2/2\right]}{1 + \prod_{i=1}^{n} \exp\left[-\left(x^0 - y_i^0\right)^2/2\right]} \tag{2.17}$$

$$\begin{aligned} f\left(\boldsymbol{y}^0\right) &= \frac{\bar{z}^2 + \bar{z}^1 \prod_{i=1}^{n} \exp\left[-\left(x_i^0 - y_i^0\right)^2/2\right]}{1 + \prod_{i=1}^{n} \exp\left[-\left(x_i^0 - y_i^0\right)^2/2\right]} \\ &= \alpha \bar{z}^2 + (1-\alpha)\bar{z}^1 \end{aligned} \tag{2.18}$$

Donde

$$\alpha = \frac{1}{1 + \prod_{i=1}^{n} \exp\left[-\left(x_i^0 - y_i^0\right)^2/2\right]} \tag{2.19}$$

Desde $x^0 \neq y^0$, debe haber algunos i tales que $x_i^0 \neq y_i^0$, por lo tanto, tenemos $\prod_{i=1}^{n} \exp\left[-\left(x_i^0 - y_i^0\right)^2/2\right] \neq 1$, o $\alpha \neq 1-\alpha$. Si nosotros elija $\bar{z}^1 = 0$ y $\bar{z}^2 = 1$, luego $f\left(x^0\right) = 1-\alpha \neq \alpha = f\left(\boldsymbol{y}^0\right)$.

Por lo tanto, (Y, d_∞) separa puntos en U. Finalmente, probamos que (Y, d_∞) desaparece en ningún punto de U. Al observar (2.8) y (5), simplemente elegimos todos $\bar{z}^j > 0 (j = 1, 2, \cdots, M)$; es decir, cualquier $f \in Y$ con $\bar{z}^j > 0$ sirve como el f requerido.

En resumen, usando el teorema de Stone-Weierstrass y el hecho de que Y es un conjunto de funciones continuas reales en U, hemos probado el teorema.

Aunque este es un resultado interesante, debe tenerse en cuenta que generalmente es indeseable tener que definir un conjunto separado de funciones de pertenencia para cada regla. Además, el teorema no define el número de funciones básicas o reglas requeridas para lograr la precisión deseada (dada por ϵ) - este número podría ser muy grande en algunos casos. Dado que una de las características más importantes de las reglas difusas es que los humanos deberían poder interpretarlos, una gran cantidad de reglas podrían trabajar en contra de este propósito.

2.2. Modelos difusos de Takagi – Sugeno (TS)

Este trabajo trata principalmente de un modelo difuso de Takagi-Sugeno (TS) propuesto por Takagi, Sugeno y Kang, para desarrollar un enfoque sistemático para generar reglas difusas a partir de un conjunto de datos de entrada-salida dado. En esta sección se presentará la estructura de este modelo y los paradigmas de modelado relacionados.

2.2.1. Estructura de modelos borrosos TS de primer orden y cero

El modelo TS es una combinación de un modelo lógico y matemático. Este modelo también está formado por reglas lógicas; consiste en un antecedente difuso y una función matemática como parte consecuente. Los antecedentes de las reglas difusas dividen el espacio de entrada en varias regiones difusas, mientras que las funciones consiguientes describen el comportamiento del sistema dentro de una región determinada:

$$R_j : Si\ z_1\ es\ A_{1,i}\ y \ldots y\ z_n\ es\ A_{n,j}\ entonces\ y = f_j(q_1, \ldots, q_m) \tag{2.20}$$

Donde $\mathbf{z} = [z_1, \ldots, z_n]^T$ es el vector n-dimensional de las variables antecedente, $\mathbf{z} \int \mathbf{x}$, $\mathbf{q} = [q_1, \ldots, q_m]^T$ es el vector m-dimensional de las variables consecuentes $\mathbf{q} \int \mathbf{x}$, donde $\mathbf{x}$ denota el conjunto de todas las entradas del modelo $y = f(\mathbf{x})$. $A_{i,j}(z_i)$ denota el conjunto antecedente difuso para la i-ésima entrada. Los antecedentes de reglas difusas dividen el espacio de entrada en varias regiones difusas, mientras la función consecuente $f_j(\mathbf{q})$ describe el comportamiento del sistema dentro de una región determinada. El espíritu de los sistemas de inferencia difusa se asemeja al concepto de "divide y vencerás": el antecedente de las reglas difusas divide el espacio de entrada en un número de regiones difusas locales, mientras que los consecuentes describen el comportamiento dentro de una determinada región a través de diversos componentes.

Con respecto a las funciones de pertenencia antecedente y la estructura de las re-

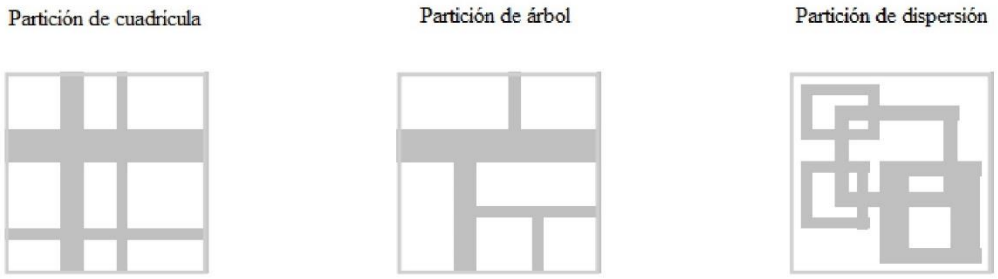


Figura 2.4: Diversos métodos para particionar el espacio de entrada.

glas, se pueden obtener tres formas típicas de partición del espacio de entrada. La figura 2.4 ilustra estas particiones en un espacio de entrada bidimensional.

- **Partición de cuadrícula.** El antecedente conjuntivo divide el espacio antecedente en una red de híper-cajas de ejes ortogonales. En este caso, el número de reglas necesarias para cubrir todo el dominio es una función exponencial de la dimensión del espacio de entrada y de los conjuntos difusos utilizados en cada variable. Este método de partición a menudo se elige al diseñar un sistema difuso que generalmente implica algunas variables de entrada. Esta estrategia de partición necesita solo un pequeño número de funciones de pertenencia para cada entrada. Sin embargo, se encuentra en problemas cuando tenemos un número moderadamente grande de entradas. Por ejemplo, un modelo difuso con diez entradas y dos funciones de pertenencia en cada entrada daría como resultado $2^{10} = 1024$ reglas difusas Si - Entonces. Este problema, generalmente conocido como la maldición de la dimensionalidad, puede aliviarse mediante otras estrategias de partición introducidas abajo.

- **Partición de árbol.** Esta partición alivia el problema del aumento exponencial del número de reglas. Sin embargo, se necesitan más funciones de pertenencia para cada entrada para definir estas regiones difusas y estas funciones de pertenencia no son interpretables ya que no tienen un significado

lingüístico claro como "pequeño".

- **Partición de dispersión.** Al cubrir un subconjunto de todo el espacio de entrada que caracteriza una región de posible ocurrencia de los vectores de entrada, se puede obtener la partición de dispersión que también puede limitar el número de reglas a un valor de cantidad razonable. En este caso, el uso de funciones de pertenencia multivariadas es el más general, ya que no hay restricción en la forma de los conjuntos difusos. Los límites entre los conjuntos difusos pueden ser arbitrariamente curvos y opacos a los ejes.

Usualmente, la función consecuente de f_j es un polinomio en las variables de entrada, pero puede ser cualquier función elegida arbitrariamente que pueda describir adecuadamente la salida del sistema dentro de la región especificada por el antecedente de la regla. Donde $f_j(\mathbf{q})$ es un polinomio de primer orden,

$$f_j = \theta_j^0 + \theta_j^1 q_1 + \cdots + \theta_j^m q_m = \sum_{l=0}^{m} q_l, \ \textit{donde} \ q_0 = 1 \tag{2.21}$$

El sistema de inferencia difusa resultante se llama modelo Takagi-Sugeno de primer orden o simplemente modelo difuso Takagi-Sugeno. Si $f_j(\mathbf{q})$ es una constante (singular difuso) $f_j = \theta_j^0$ tenemos un modelo difuso Takagi-Sugeno de orden cero o un modelo difuso que no pertenece al resto, que es un caso especial de los sistemas de inferencia difusa lingüística y el modelo difuso TS $(m = 0)$. Usando la inferencia difusa basada en la gravedad del producto-suma en una entrada dada, el resultado final del modelo difuso, y, se infiere al tomar el promedio ponderado de las funciones consiguientes como se muestra en la Figura 2.5:

$$y = \frac{\sum_{j=1}^{N_r} \beta_j(\mathbf{z}) f_j(\mathbf{q})}{\sum_{j=1}^{N_r} \beta_j(\mathbf{z})} \tag{2.22}$$

Donde los pesos, $0 \leq \beta(\mathbf{z}) \leq 1$, representa el valor de verdad general de la regla i-ésima calculada en función de los grados de pertenencia.

$$\beta_j(\mathbf{z}) = \prod_{i=1}^{n} A_{i,j}(z_i). \tag{2.23}$$

La figura 2.5 muestra el procedimiento de razonamiento difuso para un modelo difuso TS.

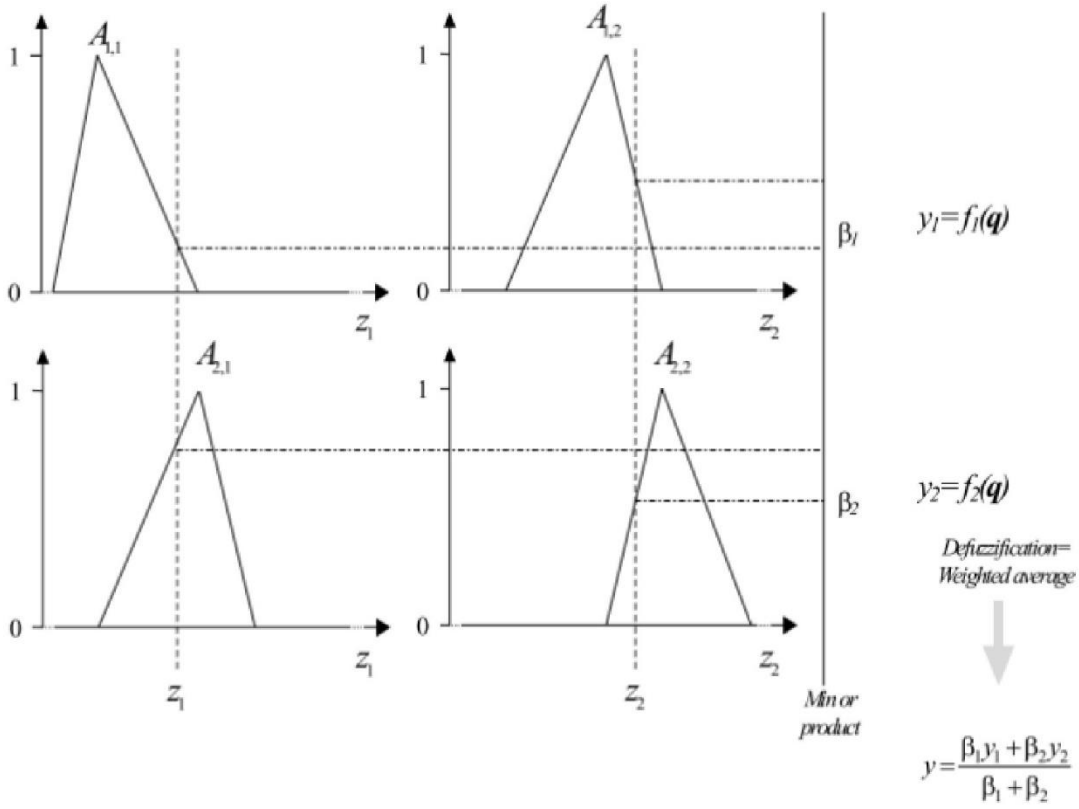


Figura 2.5: Método de inferencia del modelo difuso Takagi-Sugeno.

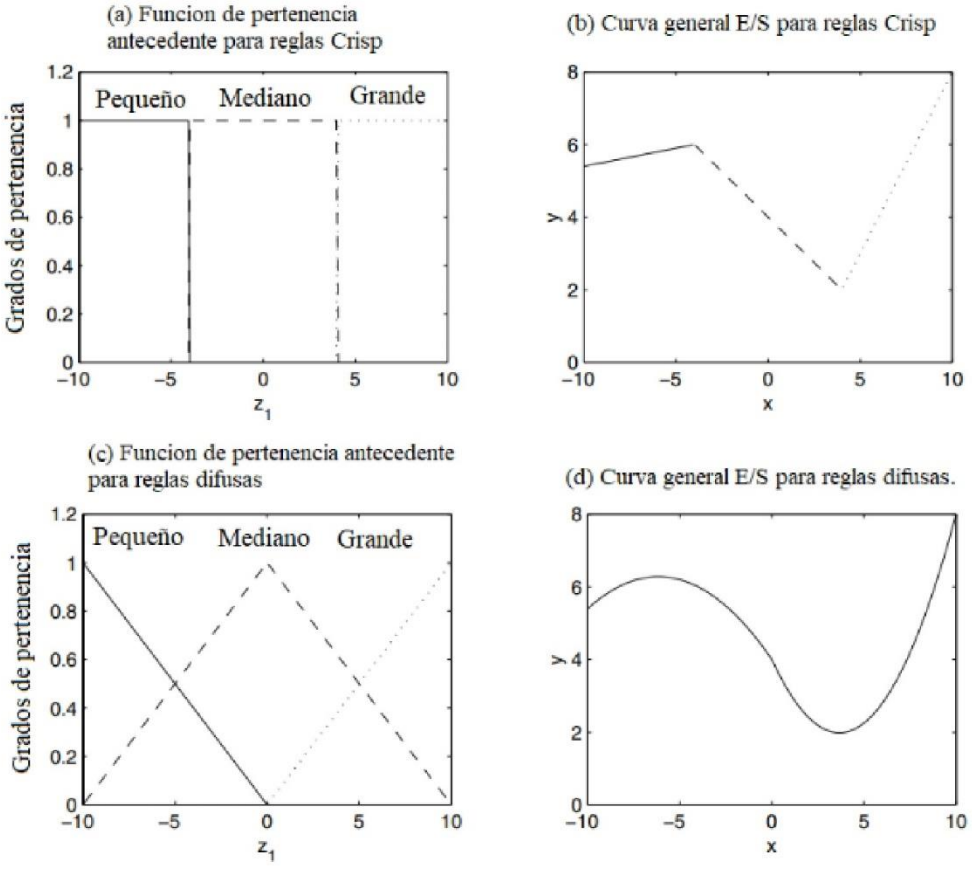


Figura 2.6: Ejemplo de un modelo difuso TS.

Como muestra la figura 2.7, las funciones de pertenencia están organizadas por una partición del tipo Ruspini manteniendo la suma de los grados de pertenencia igual a uno.

$$\sum_{i_l=1}^{i_l=M_l} A_{l,i_l}(z_l) = 1, \ l = 1, \ldots, n \tag{2.24}$$

Donde M_l representa el número de conjuntos difusos en el dominio de entrada i-ésima. Por lo tanto, las funciones de pertenencia triangular se definen por:

$$A_{l,i_l}(z_l) = \frac{z_l - a_{l,i_j-1}}{a_{l,i_l} - a_{l,i_l-1}}, \quad a_{l,i_l-1} \leq z_l < a_{l,i_l}$$

$$A_{l,i_l}(z_l) = \frac{a_{l,i_j+1} - z_1}{a_{l,i_j+1} - a_{l,i_l}}, \quad a_{l,i_l} \leq z_l < a_{l,i_l+1} \tag{2.25}$$

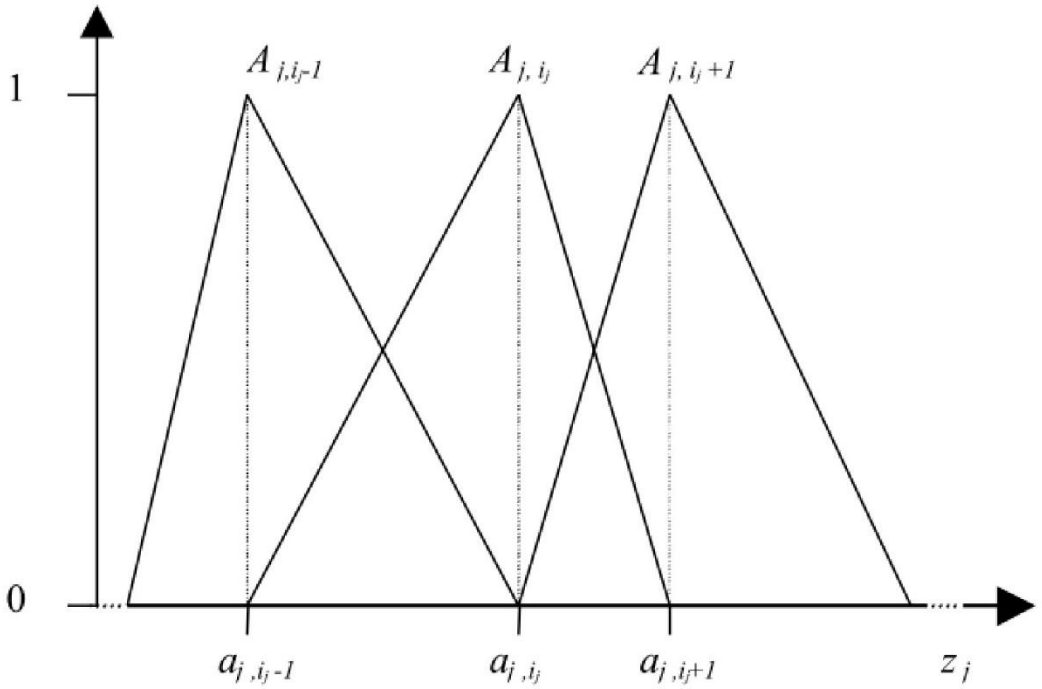


Figura 2.7: Parametrización de Ruspini de funciones de pertenencia triangular.

Donde a_{l,i_l} los núcleos de los conjuntos difusos adyacentes determinan el $(sup_{l,i_l} = a_{l,i_{l+1}} - a_{l,i_{l-1}})$ soporte de un conjunto.

$$a_{l,i_l} = core(A_{l,i_l}(z_l)) = \{A_{l,i_l} = 1\} \tag{2.26}$$

Ejemplo 5 ***(Partición de Ruspini del espacio de entrada bidimensional.)*** *En el caso multivariable, el modelo presentado obtiene una partición paralela de ejes tipo cuadricula del espacio de entrada que ayuda a obtener una base de reglas fácilmente interpretable. La Figura 2.8 ilustra dicha partición de un espacio de entrada bidimensional cuando* $n = 2$, $M_1 = 5$, $M_2 = 5$.

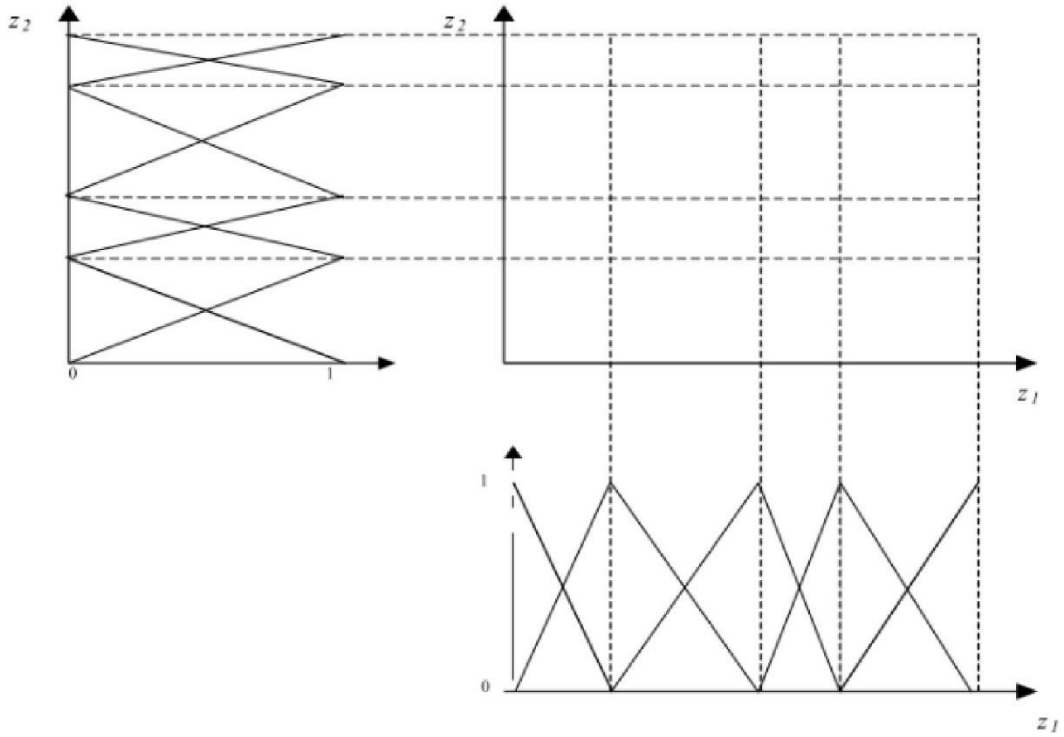


Figura 2.8: Partición del dominio de entrada (cuando $n = 2$, $M_1 = 5$, $M_2 = 5$).

A continuación, se describe el formalismo matemático de los modelos difusos TS para regresión no lineal. Considere la identificación de un sistema no lineal desconocido

$$y = f(\ x) \tag{2.27}$$

Basado en algunos datos disponibles de entrada y salida $x_k[x_{1,k}, \ldots, x_{n,k}]^T$ y y_k, respectivamente. El índice $k = 1, \ldots, N$ denota los datos muestrales individuales. A la vez puede ser difícil encontrar un modelo para describir el sistema desconocido globalmente, se logra a menudo construir modelos lineales locales alrededor de puntos operativos seleccionados. La armazón modeladora se basa al combinar modelos locales validos dentro de regiones operativas predefinidas se llama régimen operativo basado en modelamiento. En esta armazón, el modelo es gene-

ralmente dado por:

$$\hat{y} = \sum_{i=1}^{c} \phi_i(\ x)(\mathbf{a}_i^T\ x + b_i) \tag{2.28}$$

Donde $\phi_i(\mathbf{x})$ es la función de validación para el i-ésimo régimen operativo y $\theta_i = [\mathbf{a}_i^T b_i]^T$ es el parámetro vectorial del modelo lineal local correspondiente. Los regímenes operativos también pueden ser representados por conjuntos difusos en cuyo caso el modelo Takagi-Sugeno es obtenido:

$$R_i : \text{Si}\ \ x \text{ está en } A_i(\ x) \text{ entonces } \hat{y} = \mathbf{a}_i^T\ x + b_i,\ \ [w_i] \quad i = 1, \ldots, c. \tag{2.29}$$

Donde, $\mathbf{A}_i(\mathbf{x})$ es una función de pertenencia multivariante, $\mathbf{a}_i$ y b_i son parámetros del modelo lineal local, y $w_i \in [0, 1]$ es el peso de la regla. El valor de w_i esta usualmente seleccionado por el diseñador del sistema difuso para representar la convicción en la veracidad de la regla i-ésima. Cuando el conocimiento no está disponible, se usa $w_i = 1$.

La proposición antecedente " x está en $A_i(\ x)$" puede ser expresada como una combinación de proposiciones lógicas con un conjunto univariante difuso definido por los componentes individuales de x, usualmente en la siguiente forma conjunta:

$$R_i : \text{Si } x_1 \text{ es } \mathbf{A}_{i,1}(x_1) \text{ y } \ldots \text{ y } x_n \text{ está en } A_{i,n}(x_n) \text{ entonces } \hat{y} = \mathbf{a}_i^T\ x + b_i,\ \ [w_i]. \tag{2.30}$$

El grado de cumplimiento de la regla se calcula luego como el producto de los grados individuales de pertenencia y el peso de las reglas:

$$\beta_i(\ x) = w_i \mathbf{A}_i(\ x) = w_i \prod_{j=1}^{n} A_{i,j}(x_j). \tag{2.31}$$

Las reglas son agregadas usando la fórmula del promedio difuso

$$\hat{y} = \frac{\sum_{i=1}^{c} \beta_i(\mathbf{x})(\mathbf{a}_i^T\ x + b_i)}{\sum_{i=1}^{c} \beta_i(\ x)} \tag{2.32}$$

2.2.2. Paradigmas relacionados con el modelamiento.

Existen muchas estrategias de modelamiento bien conocidas o en desarrollo que pueden considerarse como un caso especial del modelo difuso presentado anteriormente. La parte restante de esta sección presenta las conexiones con estos métodos para mostrar la posible interpretación de los modelos difusos TS.

- **Modelamiento basado en regiones operativas.**

  Como la combinación de conjuntos difusos divide el espacio de entrada en varias regiones difusas y las funciones consecuentes (modelos locales) describen el comportamiento del sistema dentro de una región determinada, el modelo difuso TS puede verse como una red de modelos múltiples. La transición suave entre los regímenes operativos es manejada por el sistema difuso de manera elegante. Esta representación es atractiva, a que muchos sistemas combinan su comportamiento sin problemas en función del punto de operación.

  De (2.28) y (2.32) se puede ver que el modelo difuso TS es equivalente al modelo basado en el régimen operativo cuando la función de validez se elige como el grado normalizado de la regla de cumplimiento:

$$\phi_i(\mathbf{x}) = \frac{\beta_i(\mathbf{x})}{\sum_{i=1}^{c} \beta_i(\mathbf{x})} \tag{2.33}$$

  En este capítulo, las funciones de pertenencia Gaussiana se utiliza para representar los conjuntos difusos $A_{i,j}(x_j)$:

$$A_{i,j}(x_i) = \exp\left(-\frac{1}{2}\frac{(x_j - v_{i,j})^2}{\sigma_{i,j}^2}\right) \tag{2.34}$$

Con $v_{i,j}$ siendo el centro y $\sigma_{i,j}^2$ la varianza de la curva Gaussiana. Esta elección lleva a la siguiente formula compactada por (2.31):

$$\beta_i(\mathbf{x}) = w_i \mathbf{A}_i(\ x) = w_i \exp\left(-\frac{1}{2}(\ x - \mathbf{v}_j^x)^T (\mathbf{F}_i^{xx})^{-1} (\mathbf{x} - \mathbf{v}_j^x)\right). \tag{2.35}$$

El centro del vector es denotado por $\mathbf{v}_j^x = [v_{1,j}, \ldots, v_{n,j}]$ y $(\mathbf{F}_i^{xx})^{-1}$ es la inversa de la matriz que contienen las varianzas en su diagonal:

$$\mathbf{F}_i^{xx} = \begin{bmatrix} \sigma_{1,i}^2 & 0 & \cdots & 0 \\ 0 & \sigma_{2,i}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{n,i}^2 \end{bmatrix} \tag{2.36}$$

- **Modelos de piezas instruidas.**

  Cuando el antecedente de un modelo difuso TS de primer orden consiste en conjuntos crisp o los conjuntos difusos se definen mediante funciones de pertenencia lineal por partes, el modelo difuso resultante tiene un modelo de piezas instruidas (lineal o cuadrático). Las técnicas de modelamiento basados en modelos de piezas instruidas lineales se aplican ampliamente para el modelamiento de control relevante. Skeppstedt describe el uso de modelos locales para el modelamiento y con propósitos de control de transferencia dura de un modelo al siguiente. Portman describe una aproximación de modelos múltiples donde los modelos locales se traslapan. Estos modelos pueden ser usados eficazmente en un modelo basado en control.

  Cuando la regla consecuente es un número crisp (singleton) y el antecedente de la regla contiene funciones de pertenencia de piezas instruidas lineales, el modelo difuso resultante tiene un comportamiento de entrada-salida lineal de partes instruidas. Los conjuntos difusos multidimensionales de piezas instruidas pueden ser obtenidos por la triangulación de proposiciones características definidas en el espacio de entrada del modelo difuso. Esta técnica ya ha sido sugerida en el contexto de reducción de la comple-

jidad de los sistemas difusos. Además, la aproximación Delaunay basada en una función de empalme suave multivariante de muestras dispersas de una función desconocida ha resultado ser una herramienta efectiva para la clasificación. Recientemente, las Redes Delaunay fueron introducidas para representar modelos interpolares y controladore y la integración de conocimiento experto en estos modelos ha sido asimismo estudiada.

- **Redes B- Empalme Suave.**

  El modelo difuso presentado tipo rejilla hace aproximación polinómica por piezas instruidas del sistema no lineal a ser modelado. Los polinomios de piezas instruidas están estrechamente relacionados con los modelos de empalme suave y se han aplicado con éxito a muchos problemas de la vida real, por ejemplo, en el control experimental del pH y para la predicción de la viscosidad para un reactor de polimerización industrial. Las funciones básicas de empalme suave pueden considerarse como polinomios de piezas instruidas, donde las funciones básicas de empalme suave se definen mediante un conjunto de nodos que representan intervalos de polinomios de piezas instruidas. El orden de estos polinomios locales es definido por el orden de B-empalme suaves, denotados por k. Un nudo vectorial de un conjunto de funciones básicas de orden k es definido por:

$$\mathbf{a}_i = [a_{i,1}, a_{i,2}, \ldots, a_{i,M_i+k-1}]^T \tag{2.37}$$

  Donde M_i es el número de funciones básicas definidas sobre la i-ésima variable, y $a_{i,j}$ es el j-ésimo nodo. Las funciones básicas univariante se calculan utilizando la siguiente relación de recurrencia:

$$A_{i,j}^k(z_i) = \left(\frac{z_i - a_{i,j-k}}{a_{i,j-1} - a_{i,j-k}}\right) A_{i,j-1}^{k-1}(z_i) + \left(\frac{a_{i,j} - z_i}{a_{i,j} - a_{i,j-k+1}}\right) A_{i,j}^{k-1}(z_i) \tag{2.38}$$

$$A_{i,j}^1(z_i) = \begin{cases} 1, & si\ z_i \in [a_{i,j-1}, a_{i,j}] \\ 0, & en\ otro\ caso \end{cases} \tag{2.39}$$

Donde $A_{i,j}^k(z_i)$ es la j-ésima función básica univariante de orden k. Las funciones triangulares de pertenencia son idénticas a las funciones básicas B-empalme suave de segundo orden (2.25).

La multidimensionalidad del modelo se logra por medio del producto vectorial de las B-empalme suave univariante. Dado un conjunto de B-empalmes suaves definido sobre las variables de entrada, el B-empalme multivariante se puede definir como:

$$\beta_j^k(\ z) = \prod_{i=1}^{n} A_{i,j}^k(z_i). \tag{2.40}$$

Estos B-empalmes suaves multivariantes son idénticos a los pesos de la regla j-ésima.

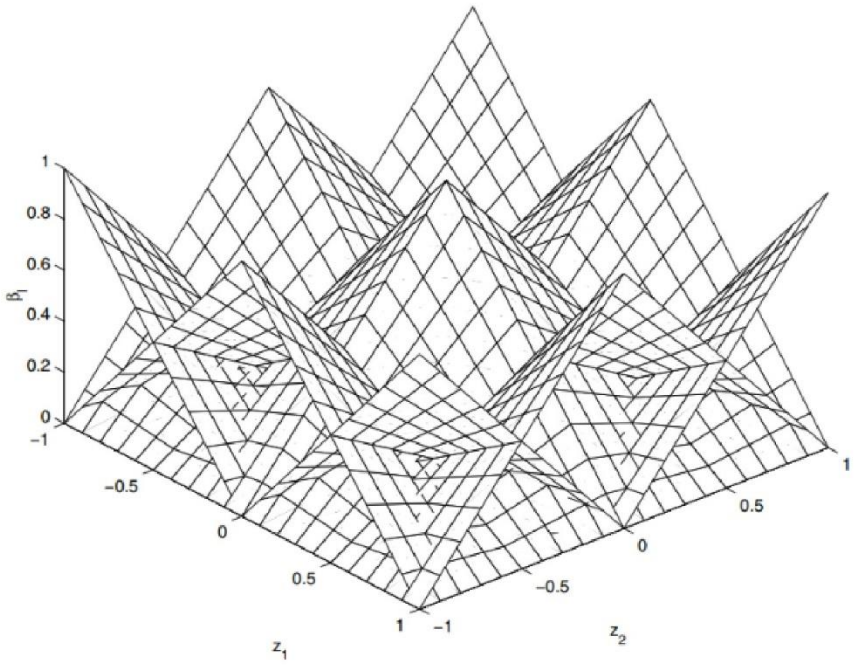


Figura 2.9: Ejemplo de funciones (de pertenencia) de empalme suave multivariante (bivariante).

La figura 2.9 muestra un ejemplo de B-empalme suave bivariante que es idéntica a los de pesos de las reglas de las nueve reglas de un sistema difuso

con tres funciones de pertenencia definidas sobre dos dominios de entrada.

- **Redes de Funciones Básicas Radiales.**

  Las redes de funciones básicas se han utilizado para la aproximación y modelado de funciones de diversas formas durante muchos años. Los métodos originales de funciones básicas radiales provienen de la teoría de interpolación, donde una función básica es asociada con cada dato puntual. Las redes de funciones básicas también han recibido atención de las redes neuronales y la comunidad de control.

  Redes de Funciones Básicas Radiales (RBFN), como propusieron en 1989 Moody y Darken, a menudo se consideran un tipo de red neuronal en la que cada unidad tiene solo un efecto local en lugar de un efecto global como en la red neuronal basada en perceptron multicapa (MPL). De manera similar a MLP, RBFN realiza una aproximación de funciones suponiendo un conjunto de N_r funciones básicas radiales (RBFN) de la siguiente manera:

$$\beta_j = \exp\left(-\sum_{i=1}^{n}\left(\frac{x_i - a_{i,j}}{\sigma_{i,j}}\right)^2\right) \tag{2.41}$$

$$y = \frac{\sum_{j=1}^{N_r}\beta_j\theta_j}{\sum_{j=1}^{N_r}\beta_j} \tag{2.42}$$

  Donde $\beta_j : \ j = 1,\ldots,N_r$ es la fuerza de disparo de la unidad j, N_r es el número de RBF, $x_i : \ i = 1,\ldots,n$ son las entradas, y es la salida, y $a_{i,j} : \ i = 1,\ldots,n; \ j = 1,\ldots,N_r$ $\sigma_{i,j} : \ i = 1,\ldots,n; \ j = 1,\ldots,N_r$ y $\theta_j : \ j = 1,\ldots,N_r$ son los parámetros libres que determinan correspondientemente la posición, el ancho y la altura de las jorobas.

  Las RBFN pueden ser entrenadas de la misma manera que las MLP, es decir, se inicializan aleatoriamente y luego se minimizan por el gradiente descendiente.

Alternativamente, la posición de los centros de las RBF y sus anchos se pueden determinar mediante un algoritmo y luego las alturas se pueden establecer mediante un tipo de algoritmo de mínimos cuadrados. Al igual que las MLP, han sido probadas para los aproximadores universales.
Jang ha señalado, bajo ciertas restricciones que la red de funciones básicas radiales (RBFN) es funcionalmente equivalente al modelo difuso TS de orden cero como

$$\beta_j = \exp\left(-\sum_{i=1}^{n}\left(\frac{x_i - a_{i,j}}{\sigma_{i,j}}\right)^2\right)$$

$$= \prod_{i=1}^{n} \underbrace{\exp\left(-\left(\frac{x_i - a_{i,j}}{\sigma_{i,j}}\right)^2\right)}_{A_{i,j}(x_i)}. \tag{2.43}$$

Hunt ha desarrollado una red de funciones básicas radiales generalizada (GBFN) que es similar al modelo difuso TS de primer orden. Estos modelos son idénticos a los modelos difusos TS bajo las siguientes condiciones:

- El número de unidades de funciones básicas es igual al número de reglas difusas si-entonces.
- Tanto la red de funciones básicas como el sistema de inferencia difusa utilizan el mismo método para calcular sus salidas del conjunto completo.

Por tanto, el modelo presentado en la Sección 3.2.1 se puede ver como RBFN si $q = 0$ y GBFN si $q \neq 0$, con funciones básicas lineales de piezas instruidas. La realización de esos modelos difusos son muy similares a las funciones aproximadoras RBFN; también significa que los métodos que se han desarrollado en el control difuso, tal como son analizados en este trabajo, pueden ser aplicados en el control neural.

2.2.3. Modelos Difusos TS para Regresión No Lineal.

La identificación difusa es una herramienta eficaz para la aproximación de sistemas no lineales con base en los datos medidos. Entre las diferentes técnicas de modelamiento difuso, el modelo Takagi-Sugeno (TS) ha atraído la mayor atención. Este modelo consta de reglas Si-Entonces con antecedentes difusos y funciones matemáticas en la parte consecuente. Los antecedentes difusos dividen en particiones el espacio de entrada en un número de regiones difusas, mientras que las funciones consecuentes describen el comportamiento del sistema en estas regiones.

La construcción de un modelo TS generalmente se realiza en dos pasos. En el primer paso, se determinan los conjuntos difusos (funciones de pertenencia) en las reglas antecedentes. Esto se puede hacer manualmente, utilizando el conocimiento del proceso, o mediante algunos datos técnicos. En el segundo paso, se estiman los parámetros de las funciones consecuentes. Como estas funciones se eligen generalmente para que sean lineales en sus parámetros, se pueden aplicar métodos estándar de mínimos cuadrados lineales.

El cuello de botella del procedimiento de construcción es la identificación de las funciones antecedentes de pertenencia que es un problema de optimización no lineal. Por lo general, se utilizan técnicas de optimización neuro-difusa de gradiente descendente, con todos los inconvenientes inherentes de los métodos de gradiente descendente: (1) La optimización es sensible a la elección de los parámetros iniciales y por lo tanto, puede fácilmente atascarse en los mínimos locales; (2) el modelo obtenido tiene propiedades deficientes de generalización; (3) durante el proceso de optimización, las reglas difusas pueden perder su significado inicial (es decir, validez como modelos locales del sistema en estudio). Este dificulta la interpretación a posteriori del modelo optimizado TS. Una solución alternativa son los algoritmos de optimización no lineal sin gradientes. Los algoritmos genéticos demostraron ser útiles fuera de la construcción de sistemas difusos. Desafortunadamente, los

severos requisitos computacionales limitan su aplicabilidad como una herramienta de desarrollo rápido de modelos.

El agrupamiento difuso en el espacio de producto cartesiano de las entradas y salidas es otra herramienta que se ha utilizado de forma bastante amplia para obtener las funciones de pertenencia antecedente. Las características atractivas de este enfoque son la identificación simultánea de las funciones de pertenencia antecedente junto con los modelos lineales locales consecuente y la regularización implícita. Al agrupar en el espacio del producto, se obtienen inicialmente conjuntos difusos multidimensionales, que se utilizan en el modelo directamente o después de la proyección sobre las variables antecedentes individuales. Dado que generalmente es difícil interpretar conjuntos difusos multidimensionales, se prefieren los conjuntos difusos unidimensionales proyectados.

Sin embargo, la proyección y la aproximación de las funciones de pertenencia definidas puntualmente por funciones paramétricas pueden deteriorar el desempeño del modelo.

Esto se debe a dos tipos de errores: el error de descomposición y el error de aproximación. El error de descomposición se puede reducir utilizando la proyección de vectores propios y/o ajustando las funciones de pertenencia parametrizadas. Sin embargo, este ajuste puede resultar en un sobreajuste y, por lo tanto, en una mala generalización del modelo identificado.

En este capítulo, proponemos utilizar el algoritmo de agrupamiento de Gath-Geva (GG) en lugar del método de Gustafson-Kessel ampliamente utilizado, porque con el método GG, los parámetros de las funciones de pertenencia univariadas se pueden derivar directamente de los parámetros de los clúster. Mediante una transformación lineal se derivará los parámetros de los clúster. Mediante una transformación lineal de las variables de entrada, la partición antecedente se puede capturar con precisión y no se produce ningún error de descomposición. Desafortunadamente, el modelo resultante no es transparente, ya que es difícil interpretar los

términos lingüísticos definidos en la combinación lineal de las variables de entrada. Para formar un modelo fácilmente interpretable que no se basa en variables de entrada transformadas, se presenta un nuevo algoritmo de agrupamiento basado en la identificación en la Maximización de la Expectación (EM) de la mezcla de modelos Gaussianos.

Las mezclas se utilizan como modelos de datos procedentes de varias poblaciones mixtas. El algoritmo EM se ha utilizado ampliamente para estimar los parámetros de los componentes de la mezcla. Los grupos obtenidos por agrupamiento GG son funciones Gaussianas multivariante. La optimización alterna de estos clúster es idéntica a la identificación EM de la mezcla de estos modelos Gaussianos cuando el exponente de ponderación difusa $m = 2$.

En este capítulo, se presenta un nuevo prototipo de clúster, que se puede representar fácilmente mediante un modelo difuso interpretable de Takagi-Sugeno (TS). De forma similar a otros algoritmos de agrupación difusa, el método de optimización alterna se emplea en la búsqueda de los grupos. Esta nueva técnica es demostrada sobre los problemas de predicción de MPG (Millas por galón) y otro proceso de referencia no lineal.

Los resultados obtenidos se comparan con los resultados de la literatura. Se muestra que con el algoritmo modificado de Gath-Geva no solo se obtiene un buen rendimiento de predicción, sino que también mejora la interpretabilidad del modelo.

2.2.4. Identificación del Modelo Difuso basado en el agrupamiento Gath-Geva.

Las muestras de los datos disponibles se recopilan en la matriz $\mathbf{Z}$ formada por concatenación de la matriz de datos de regresión $\mathbf{X}$ y el vector de salida $\mathbf{y}$

$$\mathbf{X} = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_N^T \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \quad \mathbf{Z}^T = [\mathbf{X}\mathbf{y}] \tag{2.44}$$

Cada observación, por tanto, es un vector columna de dimensión $n+1$.

$$\mathbf{z}_k = [x_{1,k}, \ldots, x_{n,k}, y_k]^T = [\mathbf{x}_k^T y_k]^T$$

A través del agrupamiento, el conjunto de datos $\mathbf{Z}$ se divide en c grupos. Se supone que c es conocida, basado en conocimientos previos. El resultado es una matriz de partición difusa $\mathbf{U} = [\mu_{i,k}]_{c\times N}$, cuyo elemento $\mu_{i,k}$ representa el grado de pertenencia de la observación z_k en el grupo i.

Se pueden obtener grupos de diferentes formas utilizando una definición apropiada de prototipos de grupo (por ejemplo, variedades lineales) o utilizando diferentes medidas de distancia.

El algoritmo de agrupamiento de Gustafson-Kessel (GK) se ha aplicado a menudo para identificar modelos TS. Los principales inconvenientes de este algoritmo son que solo los grupos con volumen aproximadamente iguales pueden identificarse adecuadamente y que los grupos resultantes no pueden describirse directamente mediante funciones de pertenencia paramétricas univariante.

Para evitar estos problemas, se aplica el algoritmo Gath-Geva. Dado que los volúmenes de clúster no están restringidos en este algoritmo, se pueden obtener errores de aproximación más bajos y parámetros consecuentes más relevantes que con el clúster de Gustafson-Kessel(GK). Los grupos obtenidos mediante el agrupamiento de GG pueden transformarse en funciones de pertenencia exponencial definidas en el espacio transformado linealmente de las variables de entrada.

Interpretación Probabilística del Agrupamiento Gath-Geva.

El algoritmo de agrupamiento de Gath-Geva se puede interpretar en el marco probabilístico. Se denota $p(\eta_i)$ la probabilidad incondicional del grupo (normalizada de tal manera que $\sum_{i=1}^{c} p(\eta_i) = 1$), dado por la fracción de los datos explicados, $p(\ z|\eta_i)$ es el dominio de influencia del grupo y se tomará la multivariante Gaussiana $N(\mathbf{v}_i, \mathbf{F}_i)$ en términos de su media $\mathbf{v}_i$ y matriz de covarianza $\mathbf{F}_i$. El algoritmo Gath-Geva es equivalente a la identificación de una mezcla de Gaussianos que

modelan la función de densidad de probabilidad $p(\ z|\eta)$ expandida en una suma sobre los c grupos

$$p(\mathbf{z}|\eta) = \sum_{i=1}^{c} p(\ z, \eta_i) = \sum_{i=1}^{c} p(\ z|\eta_i) p(\eta_i) \tag{2.45}$$

Donde la distribución $p(\mathbf{az}|\eta)$ generada por el i-ésimo grupo está representada por la función Gaussiana

$$p(\mathbf{z}|\eta_i) = \frac{1}{(2\pi)^{\frac{n+1}{2}}\sqrt{|\mathbf{F}_i|}} \exp\left(-\frac{1}{2}(\ z - \mathbf{v}_i)^T (\mathbf{F}_i)^{-1}(\ z - \mathbf{v}_i)\right). \tag{2.46}$$

A través del agrupamiento de GG, la función de densidad conjunta $p(\ z) = p(\ x, y)$ de la variable de respuesta y y el regresor $\mathbf{x}$ se modela como una mezcla de c funciones Gaussianas multivariante de dimensión $n+1$.

La densidad condicional $p(y|\ x)$ también es una mezcla de modelos Gaussianos. Por tanto, el problema de regresión se puede formular sobre la base de esta probabilidad como

$$\begin{aligned} y = f(\mathrm{x}) &= E[y|\mathrm{x}] \\ &= \int y p(y|\mathrm{x}) dy = \frac{\int y p(y, \mathrm{x}) dy}{p(\mathrm{x})} \\ &= \sum_{i=1}^{c} \frac{[[\mathrm{x}^T 1]\theta_i] p(\mathrm{x}|\eta_i) p(\eta_i)}{p(\mathrm{x})} = \sum_{i=1}^{c} p(\eta_i|\mathrm{x})[[\mathrm{x}^T 1]\theta_i]. \end{aligned} \tag{2.47}$$

Donde, θ_i es el vector de parámetros de los modelos locales que se obtendrán más adelante y $p(\eta_i|\mathrm{x})$ es la probabilidad de que el i-ésimo componente Gaussiano sea generado por el vector de regresión x:

$$p(\eta_i|\mathrm{x}) = \frac{\frac{p(\eta_i)}{(2\pi)^{n/2}\sqrt{|\mathbf{F}_i^{xx}|}} \exp\left(-\frac{1}{2}(\ x - \mathbf{v}_i^x)^T (\mathbf{F}_i^{xx})^{-1}(\mathrm{x} - \mathbf{v}_i^x)\right)}{\sum_{i=1}^{c} \frac{p(\eta_i)}{(2\pi)^{n/2}\sqrt{|\mathbf{F}_i^{xx}|}} \exp\left(-\frac{1}{2}(\mathrm{x} - \mathbf{v}_i^x)^T (\mathbf{F}_i^{xx})^{-1}(\mathrm{x} - \mathbf{v}_i^x)\right)} \tag{2.48}$$

Donde $\mathbf{F}_i^{xx}$ se obtiene creando particiones en la matriz de covarianza $\mathbf{F}$ de la siguiente manera

$$\mathbf{y} = \begin{bmatrix} \mathbf{F}_i^{xx} & \mathbf{F}_i^{xy} \\ \mathbf{F}_i^{yx} & \mathbf{F}_i^{yy} \end{bmatrix} \tag{2.49}$$

Donde:

- $\mathbf{F}_i^{xx}$ es la submatriz de $n \times n$ que contiene las primeras n filas y columnas de $\mathbf{F}_i$,
- $\mathbf{F}_i^{xy}$ es un vector columna de $n \times 1$ que contiene los primeros n elementos de la última columna de $\mathbf{F}_i$,
- $\mathbf{F}_i^{yx}$ es un vector fila de $n \times 1$ que contiene los primeros n elementos de la última fila de $\mathbf{F}_i$, y
- $\mathbf{F}_i^{yy}$ es el último elemento de la última fila de $\mathbf{F}_i$.

Construcción de Funciones de Pertenencia Antecedentes

El "Modelo Gaussiano de Mezcla de Regresores" definido por (2.47) y (2.48) es de hecho una especie de modelo basado en el régimen operativo (2.28) donde la función de validez se elige como Además, este modelo también es equivalente al modelo difuso TS donde se dan los pesos de la regla en (2.49) por:

$$w_i = \frac{p(\eta_i)}{(2\pi)^{n/2}\sqrt{|\mathbf{F}_i^{xx}|}} \tag{2.50}$$

Y las funciones de pertenencia son las Gaussianas definidas. Sin embargo, en este caso, $\mathbf{F}_i^{xx}$ no está necesariamente en la forma diagonal (2.36) y la descomposición de $\mathbf{A}_i(\ x)$ en los conjuntos difusos univariante $A_{i,j}(x_j)$ dados por (2.34) no es posible.

Si se requieren funciones de pertenencia univariante (para fines de interpretación), dicha descomposición es necesaria. Se pueden elegir dos enfoques diferentes. El primero es una aproximación, basada en la proyección eje-ortogonal de $\mathbf{A}_i(\ x)$. Esta aproximación típicamente introducirá algún error de descomposición, que puede, hasta cierto punto, ser compensado mediante la re-estimación por mínimos cuadrados globales de los parámetros consecuente.

De esta manera, sin embargo, se pierde la interpretación de los modelos lineales locales, ya que los consecuentes de la regla ya no son linealizaciones locales del sistema no lineal. El segundo enfoque es exacto, basado en la proyección de los vectores propios, también llamado enfoque de dominio de entrada transformado. Se denota $\lambda_{i,j}$ y $\mathbf{t}_{i,j}$, $j = 1., \ldots, n$, los valores propios y los vectores unitarios propios de $\mathbf{F}_i^{xx}$, respectivamente. A través de la proyección de vectores propios, se obtiene el siguiente modelo difuso en el dominio de entrada transformado:

$$\begin{aligned} R_i: \text{ Si } \tilde{x}_{i,1} \text{ está en } A_{i,1}(\tilde{x}_{i,1}) \text{ y } \ldots \text{ y } \tilde{x}_{i,n} \text{ está en } A_{i,n}(\tilde{x}_{i,n}) \\ \text{Entonces } \hat{y} = \mathbf{a}_i^T\ x + b_i \end{aligned} \tag{2.51}$$

Donde $\tilde{x}_{i,j} = \mathbf{t}_{i,j}^T\ x$ son las variables de entrada transformadas. Las funciones de pertenencia Gaussianas son dadas por:

$$A_{i,j}(\tilde{x}_{i,j}) = \exp\left(-\frac{1}{2}\frac{(\tilde{x}_{i,j} - \tilde{v}_{i,j})^2}{\tilde{\sigma}_{i,j}^2}\right) \tag{2.52}$$

Con los centros de grupo $\tilde{v}_{i,j} = \mathbf{t}_{i,j}^T \mathbf{v}_i^x$ y las varianzas $\tilde{\sigma}_{i,j}^2 = \lambda_{i,j}^2$.

Estimación de los Parámetros Consecuentes

Se presentan dos métodos de mínimos cuadrados para la estimación de los parámetros en los modelos consecuentes lineales locales: mínimos cuadrados totales ponderados y mínimos cuadrados ordinarios ponderados.

- **Estimación de Mínimos Cuadrados Ordinarios**

  El método ordinario de mínimos cuadrados ponderados se puede aplicar para estimar los parámetros consecuentes en cada regla separadamente, minimizando el siguiente criterio:

  $$\min_{\boldsymbol{\theta}_i} \frac{1}{N}(\mathbf{y} - \mathbf{X}_e\boldsymbol{\theta}_i)^T \Phi_i (\mathbf{y} - \mathbf{X}_e\boldsymbol{\theta}_i) \tag{2.53}$$

  Donde $\mathbf{X}_e = [\mathbf{X}\ \mathbf{1}]$ es la matriz de Regresores extendida por una columna unitaria y Φ_i es una matriz teniendo los grados de pertenencia en su diagonal

principal:

$$\Phi_i = \begin{bmatrix} \mu_{i,1}^2 & 0 & \cdots & 0 \\ 0 & \mu_{i,2}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mu_{i,N}^2 \end{bmatrix} \tag{2.54}$$

La estimación de mínimos cuadrados ponderados de los parámetros consecuentes están dados por:

$$\Phi_i = (\mathbf{X}_e^T \Phi_i \mathbf{X}_e)^{-1} \mathbf{X}_e^T \Phi_i \mathbf{y}. \tag{2.55}$$

Donde $\mu_{i,k}$ se obtiene mediante el algoritmo de agrupamiento de Gath-Geva, la matriz de covarianza se puede utilizar directamente para obtener la estimación en lugar de (2.55):

$$\begin{aligned} \mathbf{a}_i &= (\mathbf{F}^{xx})^{-1} \mathbf{F}^{xy}, \\ b_i &= v_i^y - \mathbf{a}_i^T \mathbf{v}_i^x. \end{aligned} \tag{2.56}$$

Esto se deriva directamente de las propiedades de la estimación por mínimos cuadrados.

- **Estimación de mínimos Cuadrados Totales.**

  Dado que los grupos se aproximan localmente a la superficie de regresión, son subespacios lineales de dimensión n del espacio de regresión de dimensión $(n+1)$. En consecuencia, el valor propio más pequeño de la matriz de covarianza de i-ésimo clúster $\mathbf{F}_i$ es típicamente en órdenes de magnitud menor que los valores propios restantes. El correspondiente valor propio $\mathbf{u}_i$ es entonces el vector normal al hiperplano atravesado por los vectores propios restantes del grupo:

$$u_i^T (z - v_i) = 0 \tag{2.57}$$

  De manera similar el vector de observación $z = [\, x^T y]^T$, el vector prototipo que es particionado como $\mathbf{v}_i = [(\mathbf{v}_i^x)^T v_i^y]$, es decir, en un vector $\mathbf{v}_x$ correspondiente al regresor x, y un escalar v_i^y correspondiente a la salida y.

El vector propio se particiona de la misma manera, $\mathbf{u}_i = [(\mathbf{u}_i^x)^T u_i^y]^T$. Utilizando estos vectores particionados (2.57) puede ser escrito como

$$[(\mathbf{u}_i^x)^T u_i^y] \left([\, x^T y] - \left[(\mathbf{v}_i^x)^T v_i^y \right] \right)^T = 0 \tag{2.58}$$

A partir del cual se pueden obtener los parámetros del hiperplano definidos por el grupo

$$y = \underbrace{\frac{-1}{u_i^y}(\mathbf{u}_i^x)^T}_{\mathbf{a}_i^T} \; x + \underbrace{\frac{1}{u_i^y}(\mathbf{u}_i)^T \mathbf{v}_i}_{b_i} . \tag{2.59}$$

Aunque los parámetros se han derivado de la interpretación geométrica de los conglomerados, se puede mostrar que (2.59) es equivalente a la estimación de mínimos cuadrados totales ponderados de los parámetros consecuentes, donde cada punto de datos se pondera por el grado de pertenencia correspondiente.

El algoritmo TLS debe utilizarse cuando hay errores en las variables de entrada. Sin embargo, tenga en cuenta que el algoritmo TLS no minimiza la media de los cuadrados de los errores de predicción del modelo, a diferencia del algoritmo de mínimos cuadrados ordinario.

Además, si las variables de entrada del modelo localmente están fuertemente correlacionadas, el vector propio más pequeño no define un hiperplano relacionado con el problema de regresión; más bien refleja la dependencia de las variables de entrada.

2.2.5. Agrupamiento Modificado Gath-Geva

El principal inconveniente de la construcción de modelos difusos interpretables de Takagi-Sugeno a través del agrupamiento es que los grupos son generalmente ejes oblicuos en lugar de ejes paralelos (la matriz de covarianza difusa $\mathbf{F}^{xx}$ tiene elementos fuera de la diagonal distintos de cero) y consecuentemente un error de descomposición es hecho en su proyección. Para evadir este problema,

proponemos en esta sección un nuevo método de agrupamiento difuso.

Agrupamiento Difuso basado en la Maximización de Expectativas para la Regresión.

Cada grupo se describe mediante una distribución de entrada, un modelo local y una distribución de salida:

$$\begin{aligned} p(\ x,y) &= \sum_{i=1}^{c} p(\ x,y,\eta_i) = \sum_{i=1}^{c} p(\ x,y|\eta_i)p(\eta_i) \\ &= \sum_{i=1}^{c} p(y|\ x,\eta_i)p(\ x|\eta_i)p(\eta_i). \end{aligned} \tag{2.60}$$

La distribución de entrada, parametrizada como una Gaussiana incondicional, define el dominio de influencia de los grupos similarmente para las funciones de pertenencia multivariante (2.35)

$$p(\ x|\eta_i) = \frac{1}{(2\pi)^{n/2}\sqrt{|\mathbf{F}_i^{xx}|}} \exp\left(-\frac{1}{2}(\ x-\mathbf{v}_i^x)^T(\mathbf{F}_i^{xx})^{-1}(\ x-\mathbf{v}_i^x)\right). \tag{2.61}$$

La distribución de salida es

$$p(y|\ x,\eta_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(y-\ x^T\theta_i)^T(y-\ x^T\theta_i)}{2\sigma_i^2}\right). \tag{2.62}$$

Cuando la transparencia e interpretabilidad del modelo es importante, la matriz de covarianza del grupo $\mathbf{F}_i^{xx}$ puede reducirse a sus elementos diagonales similarmente a la versión simplificada de ejes paralelos del algoritmo de agrupamiento de Gath-Geva:

$$p(\ x_k|\eta_i) = \prod_{j=1}^{n} \frac{1}{\sqrt{2\pi\sigma_{i,j}^2}} \exp\left(-\frac{1}{2}\frac{(x_{j,k}-v_{i,j})^2}{\sigma_{i,j}^2}\right). \tag{2.63}$$

La identificación del modelo significa la determinación de los parámetros del clúster: $p(\eta_i)$, $\mathbf{v}_i^x$, $\mathbf{F}_i^{xx}$, θ_i, σ_i. A continuación, se presenta la identificación de maximización de expectativas (EM) del modelo, seguida de una reformulación del algoritmo en forma de agrupamiento difuso.

Las bases de EM son las siguientes. Suponga que conocemos algunos valores observados de una variable aleatoria z y deseamos modelos de densidad de z utilizando un modelo parametrizado por η. El algoritmo EM obtiene una $\hat{\eta}$ estimada que maximiza la función de verosimilitud $\mathcal{L}(\eta) = p(z|\eta)$ iterando sobre los siguientes dos pasos:

- **Paso – E**. En este paso, se supone que los parámetros actuales del grupo η_i son correctos y, basándose en ellos, se calculan las probabilidades posteriores $p(\eta_i|\ x, y)$.

  Estas probabilidades posteriores se pueden interpretar como la probabilidad de que un dato en particular haya sido generado por la distribución del grupo particular.

  Usando el teorema de Bayes, las probabilidades condicionales son:

$$p(\eta_i|\ x, y) = \frac{p(\ x, y|\eta_i)p(\eta_i)}{p(\ x, y)} = \frac{p(\ x, y|\eta_i)p(\eta_i)}{\sum_{i=1}^{c} p(\ x, y|\eta_i)p(\eta_i)}. \tag{2.64}$$

- **Paso – M**. En este paso, se supone que la distribución de datos actual es correcta y se buscan los parámetros de los grupos que maximizan la probabilidad de los datos. Las nuevas probabilidades incondicionales son:

$$p(\eta_i) = \frac{1}{N}\sum_{k=1}^{N} p(\eta_i|\ x, y). \tag{2.65}$$

  Las medias y las matrices de covarianza ponderadas se calculan mediante:

$$\mathbf{v}_i^x = \frac{\sum_{k=1}^{N} x_k p(\eta_i|\ x_k, y_k)}{\sum_{k=1}^{N} p(\eta_i|\ x_k, y_k)}, \tag{2.66}$$

$$\mathbf{F}_i^{xx} = \frac{\sum_{k=1}^{N} (\ x_k - \mathbf{v}_i^x)(\ x_k - \mathbf{v}_i^x)^T p(\eta_i|\ x_k, y_k)}{\sum_{k=1}^{N} p(\eta_i|\ x_k, y_k)}. \tag{2.67}$$

Para encontrar los parámetros maximizados de los modelos lineales locales, la derivada de la log-verosimilitud se establece igual a cero:

$$0 = \frac{\partial}{\partial \theta_i} \ln \prod_{k=1}^{N} p(\mathrm{x}_k, y_k) = \sum_{k=1}^{N} \frac{\partial}{\partial \theta_i} \ln p(\mathrm{x}_k, y_k)$$
$$= \frac{1}{N p(\eta_i)} \sum_{k=1}^{N} p(\eta_i|\mathrm{x}, y)(y_k - f_i(\mathrm{x}_k, \theta_i)) \frac{\partial f_i(\mathrm{x}_k, \theta_i)}{\partial \theta_i}. \tag{2.68}$$

Aquí, $f_i(\ x_k, \theta_i)$ representa los modelos consecuentes locales, $f_i(\ x_k, \theta_i) = \mathbf{a}_i^T\ x_k + b_i$.

La ecuación anterior da como resultado la identificación por mínimos cuadrados ponderados de los modelos lineales locales (2.55) con la matriz de ponderación

$$\mathbf{\Phi}_j = \begin{bmatrix} p(\eta_i|\ x_1, y_1) & 0 & \cdots & 0 \\ 0 & p(\eta_i|\ x_2, y_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & p(\eta_i|\ x_N, y_N) \end{bmatrix} \tag{2.69}$$

Finalmente, se calculan las derivadas estándar σ_i. Estas derivadas estándar son parámetros de las funciones de distribución $p(y|\ x, \eta_i)$ definidas por (2.62).

$$\sigma_i^2 = \frac{\sum_{k=1}^{N} (y_k - f_i(\ x_k, \theta_i))^T (y_k - f_i(\ x_k, \theta_i)) p(\eta_i|\ x_k, y_k)}{N p(\eta_i)}. \tag{2.70}$$

Agrupamiento Modificado Gath-Geva para la Identificación de Modelos Difusos TS

En esta sección, el algoritmo EM se reformula para proporcionar un algoritmo fácilmente implementable, similar al agrupamiento Gath-Geva, para la identificación de modelos difusos TS que no utilizan dominios de entrada transformados. Ver algoritmo 3.3.1.

Tenga en cuenta que la medida de distancia (2.75) consta de dos términos. El primero es la distancia entre los centros de los grupos y x, mientras que el segundo cuantifica el desempeño de los modelos locales lineales.

Ejemplo 6 ***(Predicción de MPG de Automóvil - Un estudio comparativo entre técnicas basadas en agrupamiento.)***

El ejemplo que se considera es la referencia de predicción de MPG de automóvil (millas por galón). Se utilizaron y compararon los siguientes métodos:

1. *GG –TLS: Agrupamiento Gath-Geva con estimación por mínimos cuadrados totales de los parámetros consecuentes.*

2. *GG – LS: Agrupamiento Gath-Geva con estimación por mínimos cuadrados ponderados ordinarios de los parámetros consecuentes.*

3. *EM – TI: El método presentado con variables de entrada transformadas.*

4. *EM –NI: El método presentado con variables de entrada original.*

Como algunos métodos de agrupamiento son sensibles a las diferencias en los rangos numéricos de las diferentes características, los datos se pueden normalizar a una media cero y una varianza unitaria:

$$\tilde{z}_{j,k} = \frac{z_{j,k} - \overline{z}_j}{\sigma_j} \tag{2.71}$$

Donde $\tilde{z}_j$ y σ_j son la media y la varianza de la variable dada, respectivamente.
El objetivo es predecir el consumo de combustible de un automóvil sobre la base de varias características determinadas, como el peso, el año del modelo, etc. El conjunto de datos se obtuvo del Repositorio de Bases de Datos de Aprendizaje Automático y Teorías de Dominio de la UCI.

Después de eliminar las muestras con valores faltantes, el conjunto de datos se redujo a 392 entradas. Este conjunto de datos se dividió en un conjunto de entrenamiento y un conjunto de prueba, cada uno con 196 muestras.
El rendimiento de los modelos se mide por la raíz cuadrada del error de predicción cuadrático medio (RMSE):

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(y_k - \hat{y}_k)^2}.$$

El poder de aproximación de los modelos identificados se compara luego con modelos difusos con el mismo número de reglas obtenidas por la Caja de Herramientas Difusa para MATLAB®.

Algoritmo 3.3.1 (Agrupamiento Gath-Geva para Modelos Takagi-Sugeno).

Inicialización

Dados el conjunto de datos Z, c definido, elige el exponente de ponderación $m = 2$ y la tolerancia de terminación $\epsilon > 0$. Inicialice la matriz de partición de modo que (2.25), (2.26) y (2.27) se mantengan.

Repetir para $l = 1, 2, \ldots$ (l es un contador de iteraciones)

Paso 1. **Calcular los parámetros de los clústers:**

- Centros de la función de pertenencia:

$$\mathbf{v}_i^{x(l)} = \sum_{k=1}^{N} \mu_{i,k}^{(l-1)} \; x_k / \sum_{k=1}^{N} \mu_{i,k}^{(l-1)}. \tag{2.72}$$

- Desviación estándar de las funciones de pertenencia Gaussiana:

$$\sigma_{i,j}^{2(l)} = \sum_{k=1}^{N} \mu_{i,k}^{(l-1)} (x_{j,k} - v_{j,k})^2 / \sum_{k=1}^{N} \mu_{i,k}^{(l-1)}. \tag{2.73}$$

- Parámetros de los modelos locales:

$$\theta_i = (\mathbf{X}_e^T \mathbf{\Phi}_i \mathbf{X}_e)^{-1} \mathbf{X}_e^T \mathbf{\Phi}_i \mathbf{y}, \tag{2.74}$$

  Donde los pesos se recogen en la matriz $\mathbf{\Phi}_i$ dada por (2.54).

- Probabilidades a priori de los conglomerados:

- Pesos de las reglas:

$$w_i = \prod_{j=1}^{n} \frac{\alpha_i}{\sqrt{2\pi\sigma_{i,j}^2}}. \tag{2.75}$$

Paso 2. Calcular la medida de la distancia $D_{i,k}^2$:

$$\frac{1}{D_{i,k}^2} = \prod_{j=1}^{n} \frac{\alpha_i}{\sqrt{2\pi\sigma i,j^2}} \exp\left(-\frac{1}{2}\frac{(x_{j,k}-v_{i,j})^2}{\sigma_{i,j}^2}\right).$$

$$= \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(y_k - f_i(\mathrm{x}_k,\theta_i))^T(y_k - f_i(\mathrm{x}_k,\theta_i))}{2\sigma_i^2}\right). \tag{2.76}$$

Paso 3. Actualizar la matriz de particiones

$$\mu_{i,k}^{(l)} = \frac{1}{\sum_{j-1}^{c}(D_{i,k}(\ z_k,\eta_i)/D_{j,k}(\ z_k,\eta_j))^{2/(m-1)}},\ 1 \le i \le c,\ 1 \le k \le N. \tag{2.77}$$

Hasta que

$$||\mathbf{U}^{(l)} - \mathbf{U}^{l-1} < \epsilon||$$

(El modelo ANFIS) y la caja de herramienta de Identificación del Modelo Difuso (FMID) basada en el agrupamiento de Gustafson-Kessel.
Las entradas al modelo difuso son: x_1: Desplazamiento, x_2: Caballo de fuerza, x_3: Peso, x_4: Aceleración y x_5: Año del modelo.

Originalmente, había 6 funciones disponibles. El primero, el número de cilindros, se descuida aquí porque los algoritmos de agrupamiento se encuentran con problemas numéricos en características con solo una pequeña cantidad de valores discretos.
Se identificaron modelos difusos con dos, tres y cuatro reglas con el método presentado. Con el modelo de dos reglas, el método de agrupamiento presentado alcanzó valores de RMSE de 2.72 y 2.85 para los datos de entrenamiento y prueba, respectivamente, que es casi el mismo rendimiento que con los modelos de tres y cuatro reglas.

La caja de herramientas FMID da resultados muy similares: valores de RMSE de 2.67 Y 2.95 para los datos de entrenamiento y prueba. Se obtuvieron resultados considerablemente peores con el algoritmo ANFIS, que dio un modelo sobre-entrenado con el RMSE de 1.97 en los datos de entrenamiento pero de 91.35 en los datos de prueba. Estos resultados indican que el método de agrupamiento presentado tiene muy buenas propiedades de generalización.

Para una comparación adicional, también damos los resultados de un modelo de regresión lineal dado. Este modelo lineal tiene siete parámetros y seis variables de entrada (las cinco variables dadas anteriormente y el número de cilindros). Los RMSE de entrenamiento y prueba de este modelo son 3.44 y 3.45, respectivamente.

También se identificaron modelos difusos con solo dos variables de entrada, donde se tomaron las características seleccionadas, donde se propuso la siguiente estructura del modelo:

$$MPG = f(Peso, Ao) \tag{2.78}$$

Como los modelos Gath-Geva y EM-TI capturan correlaciones entre las variables de entrada, el modelo difuso de TS extraído de los grupos debe utilizar funciones de pertenencia antecedente multivariante:

$$R_i : \text{Si } \ x \text{ está en } A_i \text{ Entonces } \hat{y} = \mathbf{a}_i^T \ x + b_i$$

O variables de entrada transformadas:

$$R_i : \text{Si } \mathbf{t}_{i,1}^T \ x \text{ está en } A_{i,1} \text{ y } \mathbf{t}_{i,2}^T \ x \text{ está en } A_{i,2} \text{ Entonces } \hat{y} = \mathbf{a}_i^T \ x + b_i$$

Donde $i = 1, \ldots, c, \hat{y}$ es la MPG estimada y $x^T = [Peso, \text{Año}]$. Estos modelos no pueden ser fácilmente analizados, interpretados y validados por expertos humanos, porque los conjuntos difusos (términos lingüísticos) se definen en un espacio multidimensional o linealmente transformado. Sin embargo, el método EM-NI

presentado (agrupamiento modificado Gath-Geva) da como resultado las reglas estándar con las variables antecedentes originales en la forma conjuntiva:

$$R_i : \text{Si Peso está en } A_{i,1} \text{ y Año está en } A_{i,2} \text{ Entonces } \hat{y} = a_{i,1}Peso + a_{i,2}\text{Año} + b_i \ [w_i]. \tag{2.79}$$

La Tabla 2.1 compara el rendimiento de predicción de los modelos obtenidos. Entre los cuatro enfoques presentados, solo la identificación de mínimos cuadrados totales es sensible a la normalización de los datos. Así, en la Tabla 2.1 GG-TLS-N denota los resultados obtenidos al realizar la identificación con el uso de datos normalizados.

Métodos	2reglas (entreno)	2 reglas (test)	4 reglas (entreno)	4 reglas (test)
GG-TLS	3.34	3.43	5.58	5.71
GG-TLS-N	3.25	3.57	3.43	4.00
GG-LS	2.97	2.97	2.77	2.95
EM-TI	2.97	2.95	2.62	2.93
EM-NI	2.97	2.95	2.90	2.95
ANFIS	2.67	2.95	2.37	3.05
FMID	2.96	2.98	2.84	2.94

Cuadro 2.1: Comparación del desempeño de los modelos TS identificados con dos varialbes de entrada.

Normalmente, el rendimiento del modelo en los datos de entrenamiento mejora con el creciente número de clústers, mientras que el rendimiento en los datos de evaluación mejora hasta que aparece el efecto de sobreajuste y luego comienza a degradarse (compensación sesgo-varianza). Sin embargo, cuando se aplica el método de mínimos cuadrados totales (TLS), el error de entrenamiento se hizo mayor con el aumento de la complejidad del modelo. Esto se debe a que las variables de entrada del modelo están fuertemente correlacionadas y el vector propio más pequeño no define un hiperplano relacionado con el problema de regresión, pero refleja la dependencia de las variables de entrada.

Ya para dos grupos, la diferencia entre los dos pequeños valores propios es

muy pequeña (los eigenvalores son [10,92, 37,69, $3{,}4 \times 10^5$] para el primer clúster y [$1{,}37 \times 10^5$, 37,69, 4,93] para el segundo).

El método de agrupamiento difuso presentado mostró un rendimiento ligeramente mejor que el algoritmo Gath-Geva. Dado que estos métodos identifican modelos difusos con variables de entrada transformadas, tienen un buen rendimiento debido a la partición efectiva de ejes oblicuos del dominio de entrada, que se puede ver en la Figura 2.10.
El algoritmo EM-NI produce grupos que no se rotan en el espacio de entrada (ver Figura 2.11). Estos grupos se pueden proyectar y descomponer en funciones de pertenencia fácilmente interpretables definidas en las características individuales, como se muestra en la Figura 2.12 para el modelo de dos reglas y en la Figura 2.13 para el modelo de cuatro reglas.

Sin embargo, esta restricción reduce la flexibilidad del modelo, lo que puede resultar en un peor rendimiento de predicción. Usamos EMR-TI para demostrar cuanto rendimiento hay que sacrificar por la interpretabilidad. Para este ejemplo, la diferencia del rendimiento resulta insignificante (ver Tabla 2.1).

La caja de herramientas difusa de identificación del modelo difuso (FMID) también se utilizaron para identificar modelos difusos para el problema de predicción de MPG. Como puede verse en la Tabla 2.1, el método presentado obtiene modelos difusos que tienen un buen desempeño en comparación con estas técnicas alternativas.

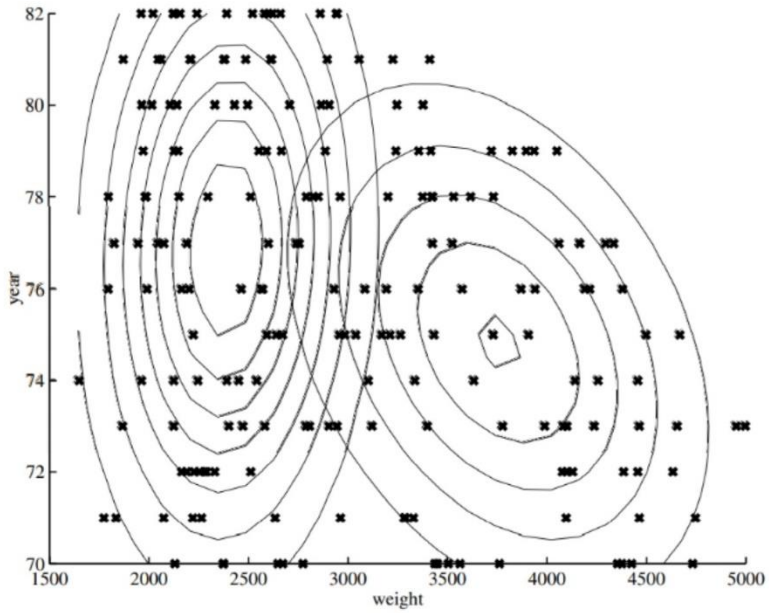


Figura 2.10: Clústers detectados por el algoritmo de agrupamiento GG.

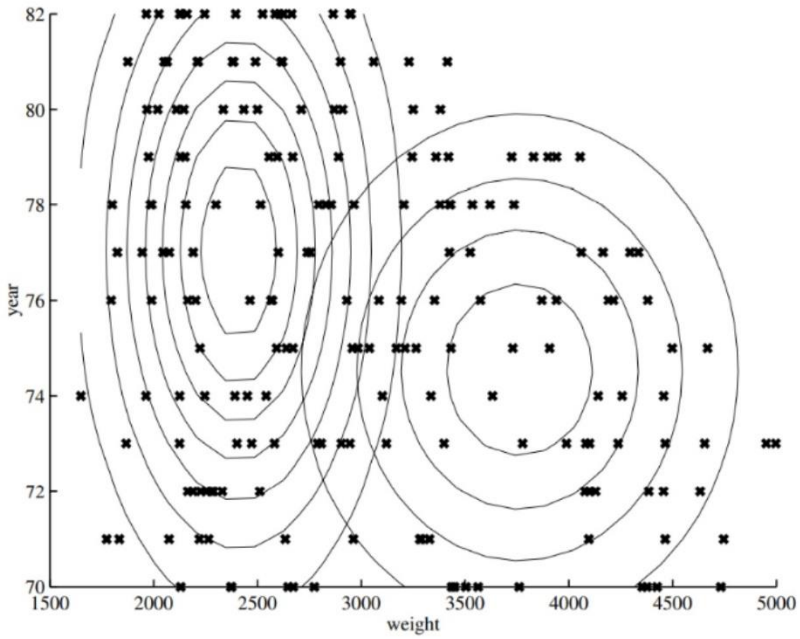


Figura 2.11: Clústers detectados por el algoritmo modificado.

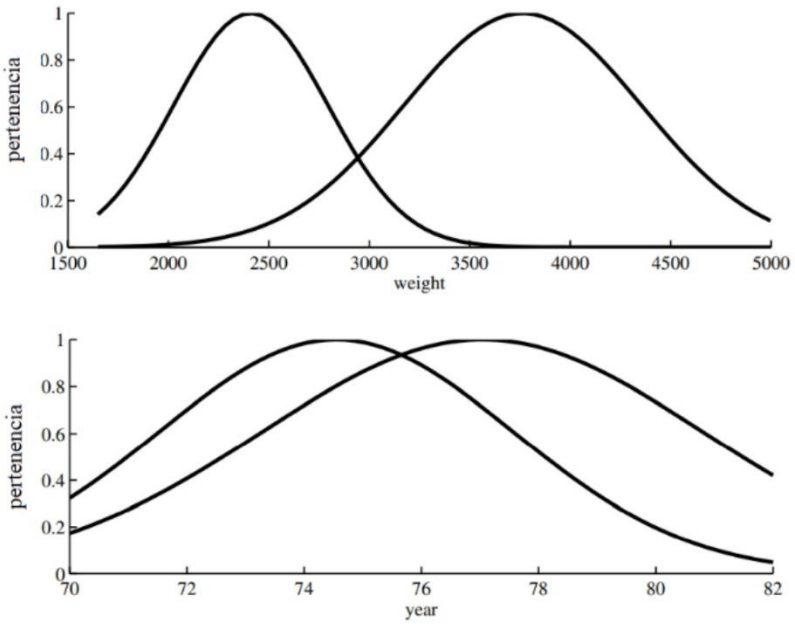


Figura 2.12: Funciones de pertenencias obtenidas.

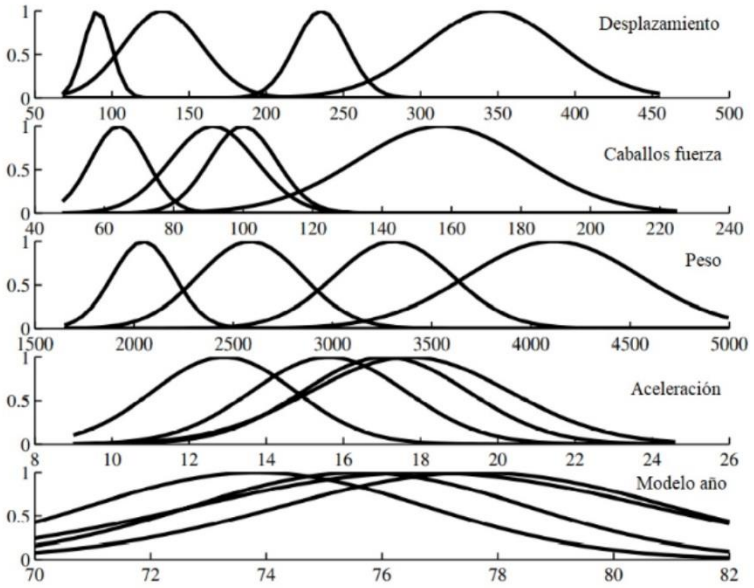


Figura 2.13: Funciones de pertenencia del modelo TS para la predicción de MPG basadas en 5 entradas.

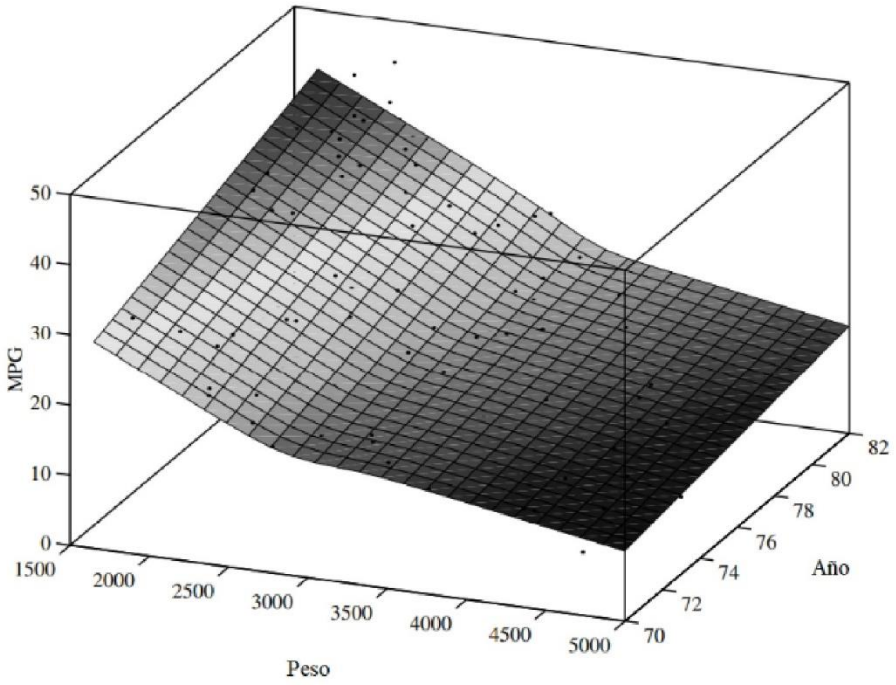


Figura 2.14: Superficie de predicción del modelo.

El modelo resultante también es bueno para la extrapolación. La superficie de predicción del modelo con dos entradas se muestra en la Figura 2.14. Si esta superficie se compara con la superficie de predicción del modelo generado por ANFIS, se puede ver que el modelo ANFIS estima falsamente un MPG más alto para autos pesados debido a la falta de datos debido a la tendencia de los fabricantes a comenzar a construir autos compactos pequeños a mediados de los 70. Como se puede ver en la Figura 2.14, el modelo EM-NI obtenido no sufre este problema.

Capítulo 3

Parte de Aplicación: Ejemplos

3.1. Ejemplo de Agrupamiento No Difuso con Datos Clínicos del Estudio IRC

Ejemplo 7 *Se ha elegido la base de datos clínicos del estudio IRC (Insuficiencia Renal Crónica) de los pobladores del bajo lempa, la cual posee las siguientes variables:*

- *Sexo*
- *IRC*
- *Edad*
- *Glucosa*
- *Creatinina*
- *Colesterol*
- *Triglicéridos*

La base de datos contiene 17 *variables de las cuales algunas son del tipo dicotómicas, por lo cual fue necesario categorizarlas. A continuación se mostraran los resultados obtenidos por medio del método de agrupamiento k-means y posteriormente se presenta el código en RStudio empleado.*

Aplicando el método k-means en R por medio de la interface de RStudio se obtuvieron los siguientes resultados:

En la imagen 3.1 presenta todos los individuos muestreados que no tenían datos faltantes, ya que k-means no puede trabajar si existe ese inconveniente.

La imagen 3.2 muestra el agrupamiento con $K = 2$ clusters, debe notarse que para el gráfico en dos dimensiones se han empleado dos de las variables más relevantes en la sintomatología de la insuficiencia renal crónica.

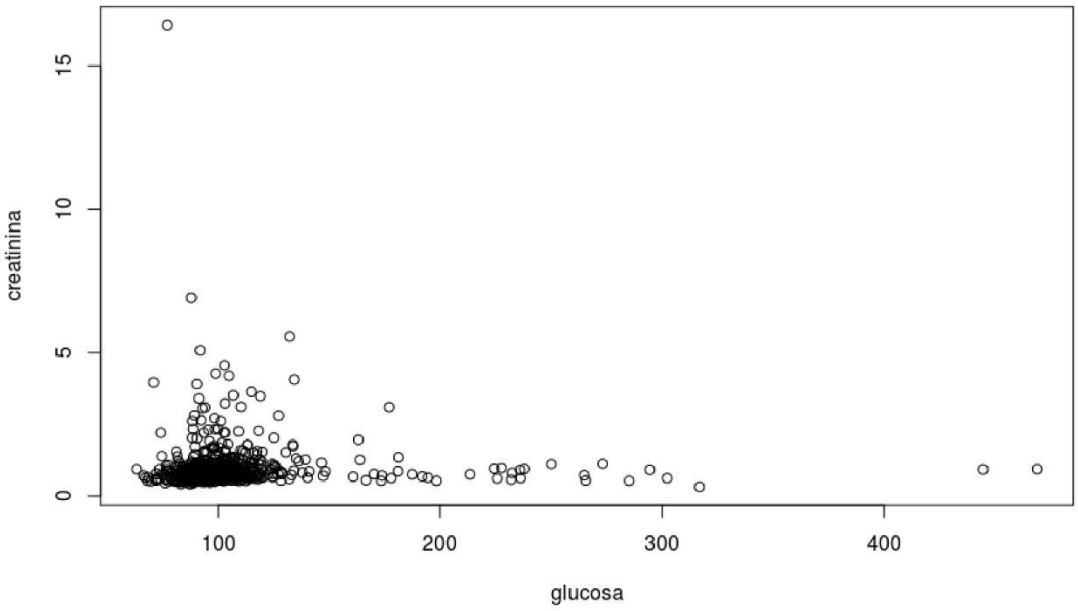


Figura 3.1: Individuos muestreados

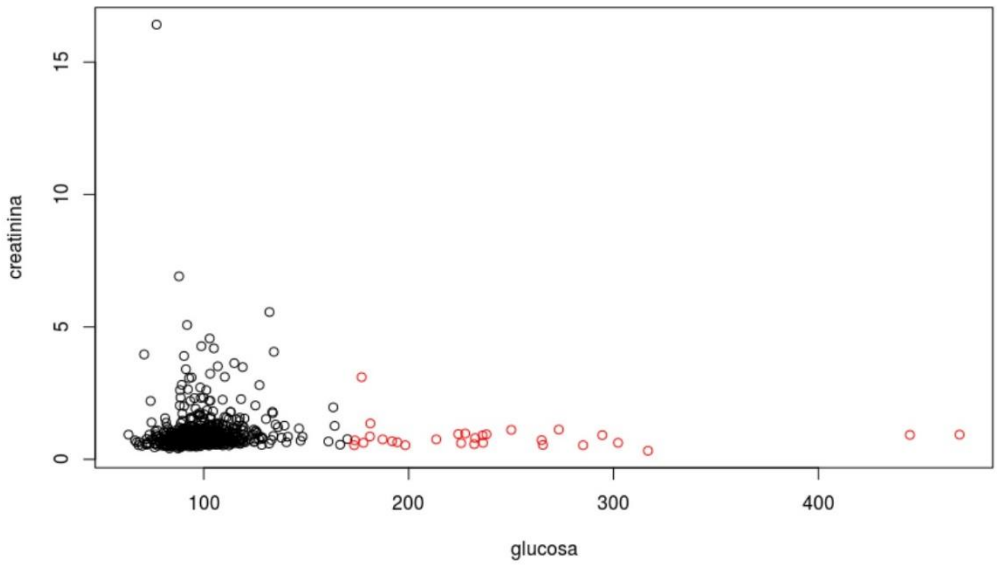


Figura 3.2: Agrupamiento con $K = 2$ clústers

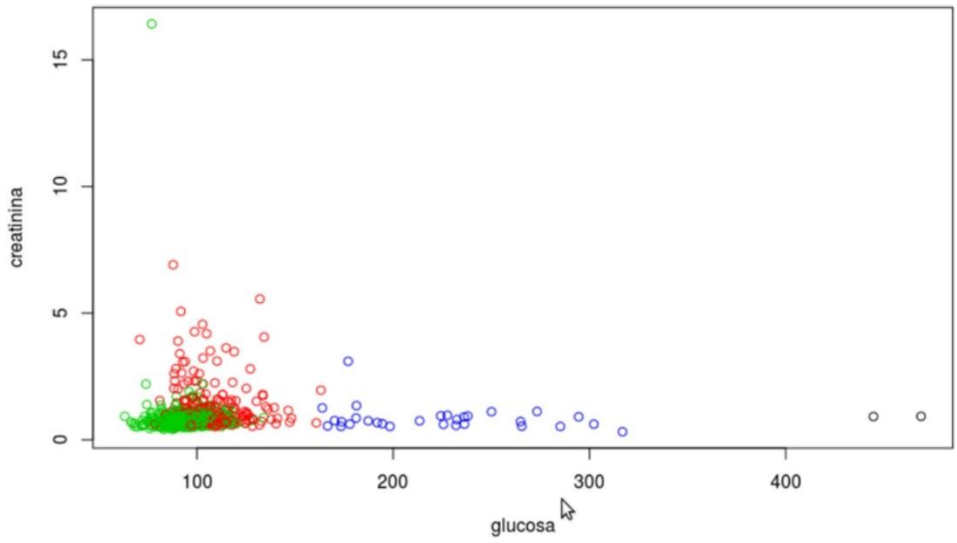


Figura 3.3: Agrupamiento con $K = 4$

La figura 3.3 muestra el agrupamiento cuando $k = 4$ clúster:

Al aumentar el número de agrupamientos podemos notar que individuos van conformando cada grupo. De aquí surge la pregunta: ¿Cuántos grupos es adecuado formar? Para responder a la pregunta se emplean diversos criterios, uno de ellos es el valor del SSE (suma de los cuadrados de los errores) que en nuestro caso empleamos SSE inter-grupos, es decir la suma de los cuadrados de los errores entre los grupos. Esta medida indica la distancia entre los grupos, la cual se desea sea la máxima entre los grupos ya que se quiere que sean lo más diferente posibles entre sí. Además, el número de cluster podría variar dependiendo la aplicación.

De manera iterada se forman los grupos con $k = 1, 2, \cdots, 10$ y se extrae dicho estadístico de los resultados obtenidos en RStudio, obteniéndose la figura 3.4.

Podemos notar que para este ejemplo apartir de cuatro clusters es una cantidad razonable para particionar nuestros datos. Cabe recalcar que el número de clusters que se formen dependera de cada problema a investigar, a la experiencia

del investigador y de los recursos que se tengan disponibles, ya sea recurso humano, económico, tecnológico, entre otros.

No necesariamente a una cantidad mayor de clusters se obtendrá una mayor calidad de resultados.

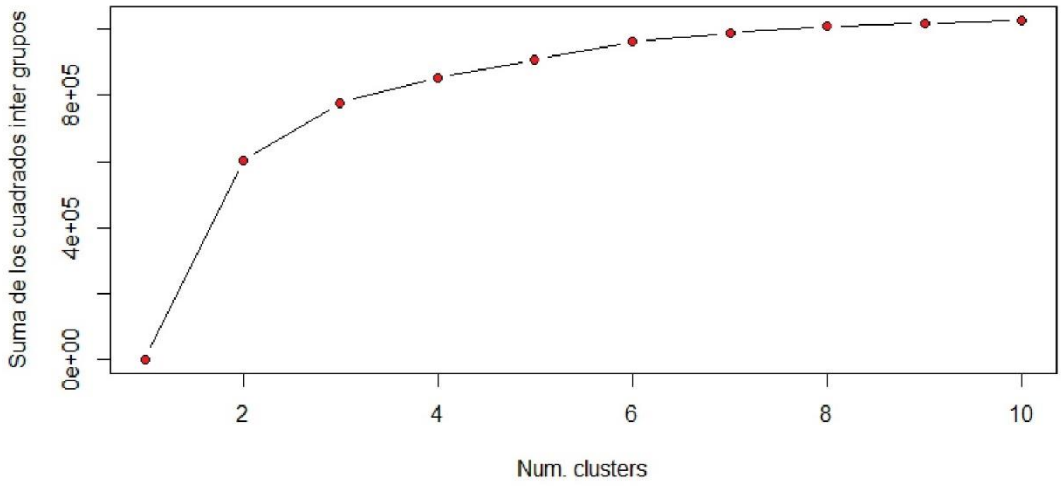


Figura 3.4: Clusters óptimo

3.2. Ejemplo del Algoritmo Difuso Gustafson-Kassel utilizando la Base de Datos Iris

Ejemplo 8 *Se utilizará la base de datos Iris, popularizada por un artículo de Fisher. Iris es quizás la base de datos más conocida que se encuentran en la literatura de reconocimiento de patrones. Esta base de datos, recolectada durante varios años por Edgar Anderson fue utilizada para demostrar que estas medidas podrían utilizarse para diferenciar entre especies de plantas iris. Contiene 3 clases de 50 casos cada una, donde cada clase se refiere a un tipo de planta iris.Los atributos son:*

- *Longitud del sépalo en cm (Sepal.Length)*
- *Ancho del sépalo en cm (Sepal.Width)*
- *Longitud del pépalo en cm (Petal.Length)*
- *Ancho del pépalo en cm (Pepal.Width)*
- *Clase (Species):*

- *Iris Setosa*
- *Iris Versicolour*
- *Iris Virginica*

Figura 3.5: Clases de plantas Iris

En la figura 3.6 se muestra: Pétalo y Sépalo de una planta. Imágen tomada de The Iris Flower Data Set Iris Abramson

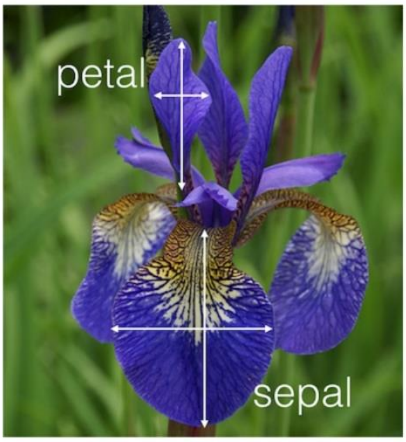


Figura 3.6: Clases de plantas Iris

Al ejecutar el código en R, se obtiene el siguiente agrupamiento, el cual se hace para tres cluster. La visualización de los resultados se muestra en la figura 3.7:

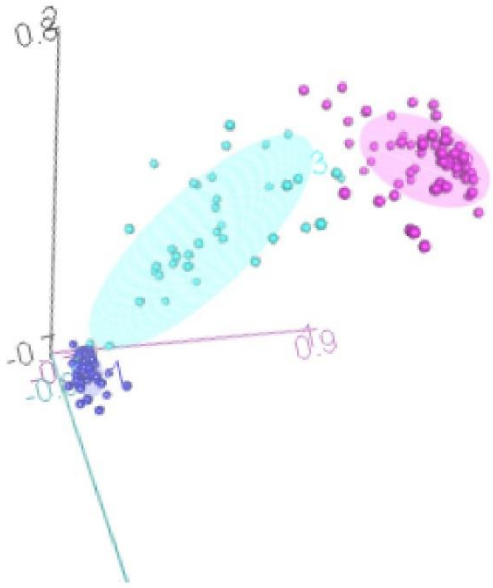


Figura 3.7: Agrupamiento por especies

Se observa que el cluster color azul esta menos disperso y conforma una figura que se asemeja a una circunferencia, por otra parte el cluster de color naranja presenta valores más dispersos en comparación al cluster azul, tanto que se confunden sus valores con los elementos que pertenencen al cluster verde y ademas se puede notar que su forma se asemeja a un elipsoide. Por otra parte el cluster de color verde presenta menos dispersión en comparación al cluster naranja, pero notemos que contiene elementos que se confunden con los elementos del cluster naranja.

Así, el cluster azul está bien definido con respecto a sus caracteristicas, en cambio los cluster naranja y verde comparten algunas características tendiendo a confundirse algunos de sus elementos, mas sin embargo, se pueden diferenciar por el color correspondiente.Notemos ademas, que el cluster de color verde conforma un elipsoide más pequeño en comparación al elipsoide conformado por el cluster naranja, esto es debido a la variabilidad de sus elementos.

3.3. Ejemplo: Modelo Difuso Takagi-Sugeno (TS).

Ejemplo 9 *Una compañía de seguros, necesita evaluar el riesgo financiero de sus clientes que requieren póliza de seguros contra accidentes automovilísticos.*

Para evaluar el riesgo financiero se toma en cuenta la edad del asegurado y su porcentaje de manejo durante el año.

Para la edad se considerará el rango de 18 a 70 años ya que es la edad permitida y se considerarán tres categorías:

- ***Joven:*** *consideraremos joven a la persona entre 18 y 30 años.*
- ***Adulto:*** *será la persona entre 20 y 50 años.*
- ***Adulto mayor:*** *a la persona cuya edad sea mayor de 40 años.*

Para el porcentaje de manejo durante el año se considerarán también tres categorías:

- ***Nivel bajo:*** *menor o igual al 20 %, es decir, menos de 7 días de conducción en cada mes.*
- ***Nivel medio:*** *del 10 % al 60 % de conducción*
- ***Nivel alto:*** *mayor al 50 % de conducción, es decir, conducir al menos 15 días en cada mes.*

Esto se expresará por medio de las siguientes funciones de pertenencia:

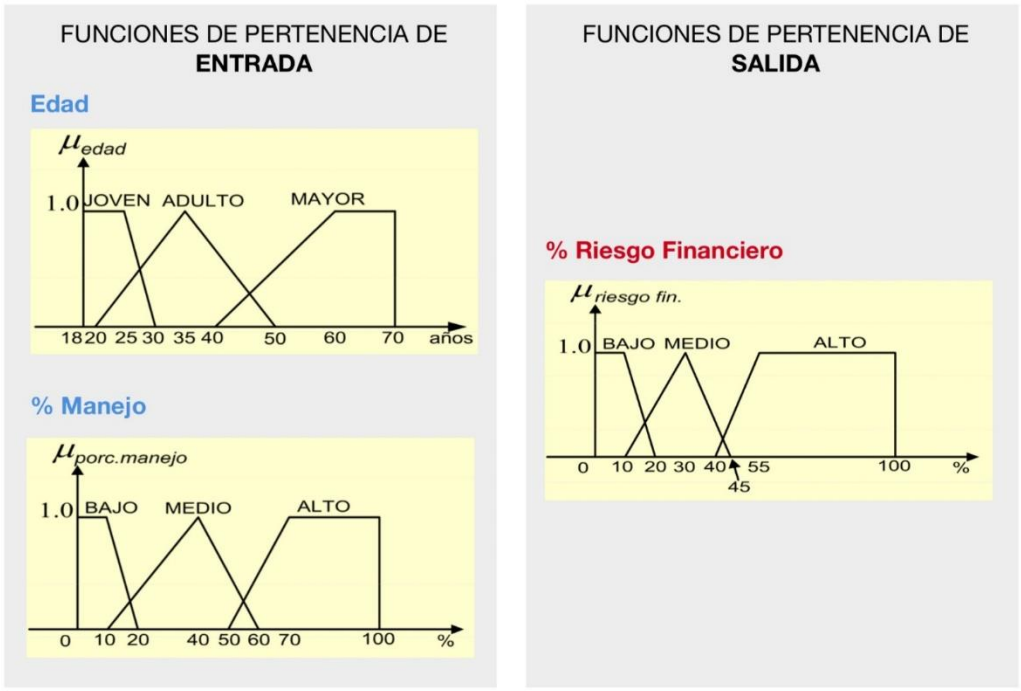


Figura 3.8: Ejemplo práctico del modelo difuso TS.

		EDAD		
		JOVEN	ADULTO	MAYOR
PORCENTAJE DE MANEJO	BAJO	MEDIO	BAJO	MEDIO
	MEDIO	ALTO	MEDIO	ALTO
	ALTO	ALTO	ALTO	ALTO

Figura 3.9: Reglas de Inferencia Difusa

Combinando estos criterios se crea la tabla de decisión o regla de inferencia difusa como se muestra en 3.9.

De manera gráfica se muestra como se implementará este ejemplo en la lógica difusa aplicando en el proceso el algoritmo de Takagi – Sugeno:

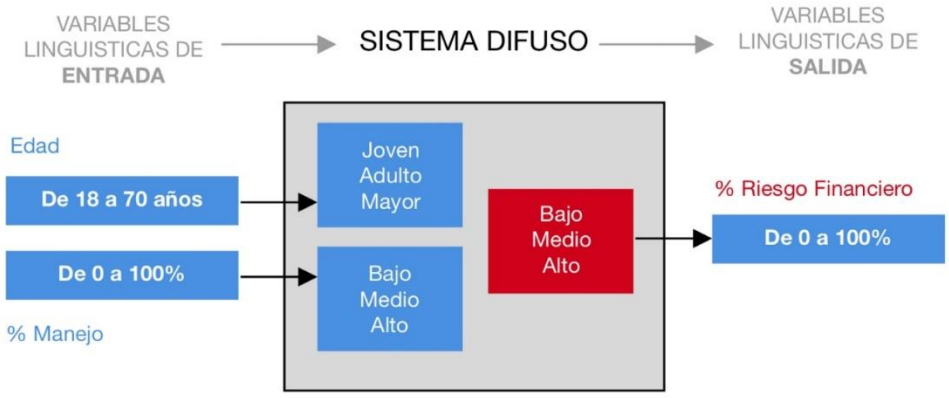


Figura 3.10: Sistema Difuso.

En la figura 3.10 se muestra como se introducen las dos variables lingüísticas de edad y porcentaje de conducción, así como los rangos que se tomarán en cuenta, luego en la salida intervendrá el algoritmo Takagi Sugeno, esto se explicará en detalle más adelante.

Mostramos en la figura 3.11 la manera en que se harpa la inferencia difusa.

En el algoritmo Takagi–Sugeno se toma el mínimo de las funciones de pertenencia, es así que para el porcentaje de riesgo financiero, este se calculará como el centroide de las figuras que intervengan en las funciones de pertenencia.

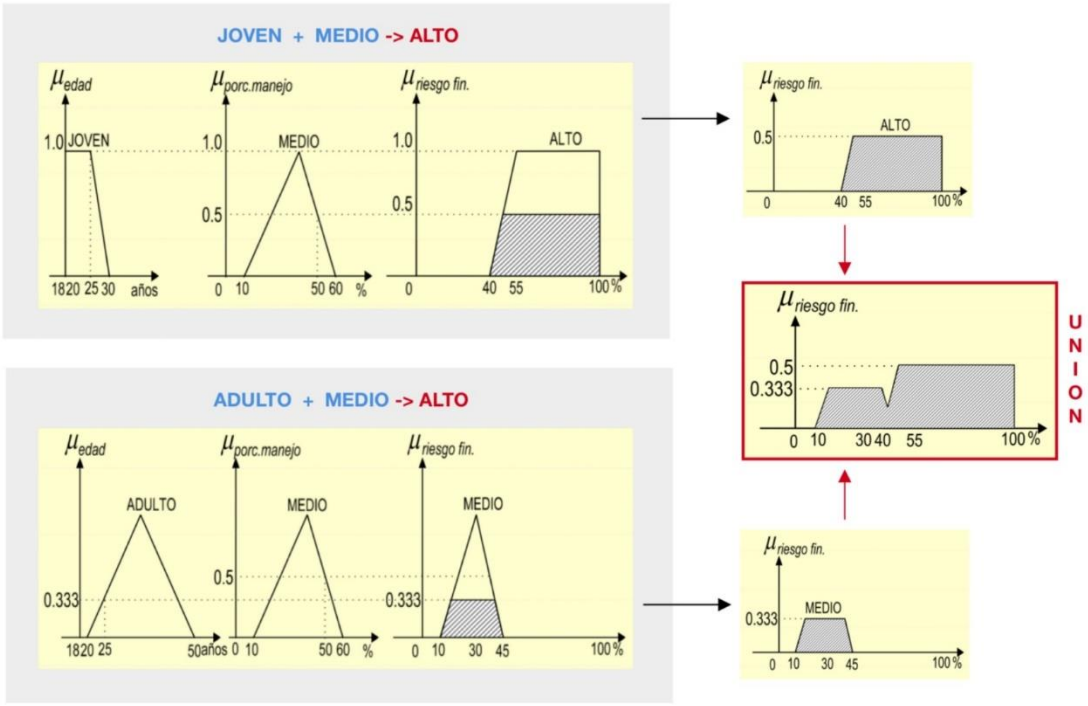


Figura 3.11: Inferencia Difusa.

Al ingresar los datos en las funciones de pertenencia se tendrá lo que se muestra en la Figura 3.12. Explicaremos esta parte suponiendo que se introducen los datos de una persona que tiene 25 años de edad y un 50 % de porcentaje de manejo. Observamos que, a partir de los datos, el porcentaje de riesgo financiero se encuentra entre las categorías medio y alto porcentaje de riesgo financiero.

Luego internamente se calculará el porcentaje de riesgo financiero que se le asignará tomando en cuenta los valores de las funciones de pertenencia y las figuras geométricas involucradas, esto para definir claramente cuál será el porcentaje de riesgo financiero.

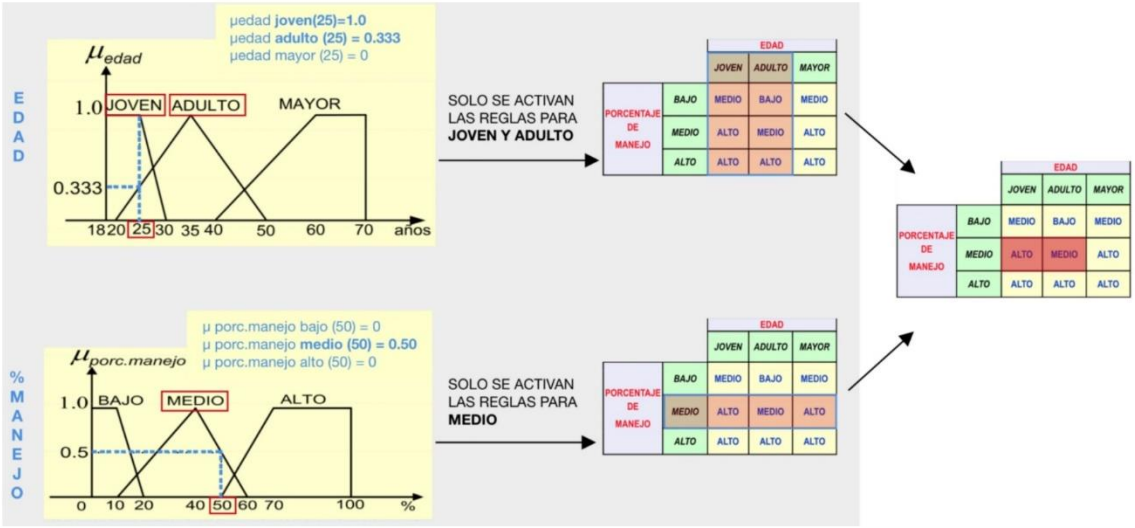
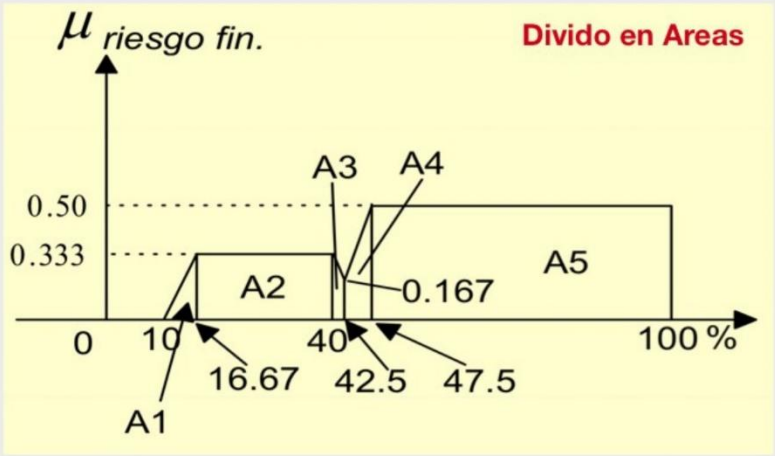


Figura 3.12: Inferencia Difusa.

Finalmente se obtiene el porcentaje de riesgo financiero al realizar los cálculos de los centroides locales y luego el centroide de la figura geométrica involucrada en el proceso.

El valor asignado del porcentaje de riesgo financiero para una persona que tiene 25 años de edad y un 50 % de porcentaje de manejo, es del 60.77 % y este corresponde a un nivel alto en el riesgo financiero.

PASO [1]



PASO [2]

Centroides Parciales

C1 = 14.447
C2 = 28.335
C3 = 40.833
C4 = 45.833
C5 = 73.75

Areas Parciales

A1 = 1.11
A2 = 7.77
A3 = 0.625
A4 = 1.67
A5 = 26.25

Area Total

A = 37.425

PASO [3]

Calculo del Centroide

$$C=\frac{1}{A}(C1A1+C2A2+C3A3+C4A4+C5A5)$$

$$C=\frac{1}{37.425}(14.447x1.11+28.335x7.77+40.833x0.625+45.833x1.67+73.75x26.25)$$

$$C=60.77$$

Figura 3.13: Método del Centroide.

De ahí que la compañía aseguradora tomará la decisión de si le otorgarán una póliza de seguro a esta persona dependiendo de las políticas que tenga dicha empresa con las personas que llegan a esa categoría, y de ser así el monto a asignarle en la póliza, así como la cuota mensual a pagar.

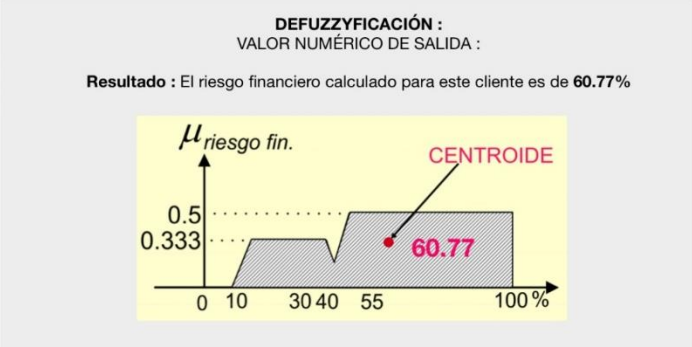


Figura 3.14: Ejemplo práctico del modelo difuso TS.

3.4. Ejemplo: Modelo Difuso SISO TS.

Ejemplo 10 *Sistema difuso Takagi-Sugeno para predecir el riesgo de propagación de Sigatoka Negra Mycosphaerella fijiensis en el cultivo de plátano, por medio del análisis de la temperatura (13-38 C) y la humedad relativa (0-100)la humedad relativa es la expresión porcentual de la cantidad de vapor de agua presente en el aire con respecto a la máxima posible para unas condiciones dadas de presión y temperatura y su medición es de 0 a 100. para las pruebas se utilizaron datos históricos del promedio mensual de temperatura y humedad relativa, desde enero de 2015 hasta junio de 2019, en Manzanillo, uno de los municipios de mayor producción de plátano del estado de Colima,Mexico.*

México es uno de los principales países productores y exportadores de plátano. Sin embargo, su calidad se ve afectada por enfermedades tales como la sigatoka negra, que ha sido la enfermedad que genera mayores pérdidas económicas, puesto que no se ha logrado controlar debido a las condiciones climáticas en las que se propaga. Las principales variables que influyen en la aparición de la sigatoka negra son la temperatura y la humedad relativa; y si no se controla, el agente patógeno puede reducir hasta en 50 % del peso del racimo y causar pérdidas de hasta 100 % de la producción.

Para la creación del SD se definieron las variables de entrada, siendo estas la temperatura y la humedad relativa, permitiendo determinar la variable de salida, riesgo de proliferación de la SN. Se especifica que las condiciones óptimas para el crecimiento de la SN se ocasionan cuando las condiciones de humedad y temperatura se encuentran entre los valores mostrados en la figura 3.15.

Tipo de sigatoka	Temperatura	Humedad
Negra	23-28 °C	80-100 %
Amarilla	Media 25 °C o mayores	90-100 %

Figura 3.15: Condiciones Climáticas de la SA y SN.

Variables de entrada.

Las variables de entrada del SD son las variables climáticas que influyen en la aparición de la SN, temperatura y humedad relativa.

Humedad Relativa.

La 3.16 muestra los valores utilizados en el SD propuesto para la humedad, con base en los rangos de medición (0-100 %) y en el rango de humedad para la SN mostrado en la figura 3.15.

FM	Rango
Poca	0 a 35 %
Media	22.5 a 77.5 %
Mucha	65 a 100 %

Figura 3.16: Rangos de Humedad Relativa.

Las FM trapezoidales tienen cuatro parámetros, donde a corresponde al ángulo inferior izquierdo, b al ángulo superior izquierdo, c al ángulo superior derecho y finalmente d al ángulo inferior derecho y se definen como:

$$MF(x) = 0 \; si \; -\infty < x \leq a$$

$$MF(x) = 1 \; si \; b \leq x \leq c$$

$$MF(x) = 0 \; si \; d \leq x < \infty$$

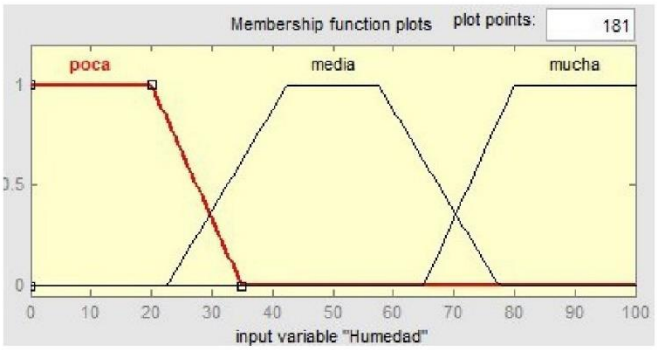


Figura 3.17: Función de membresía de la Humedad Relativa

Como se muestra en la Figura 3.17 se optó por emplear MF trapezoidales para la variable de entrada humedad relativa, debido a que son las FM que mejor se adaptan por su forma geométrica, al rango de humedad que propicia la aparición de la SN, como se puede observar en la Tabla 3.15.

Los valores de pertenencia de las funcionesde membresía de la humedad relativa fueron obtenidos con base al valor de la humedad para la SN en la Tabla 3.15 y fueron ajustados de manera empírica. Además, al utilizar MF trapezoidales se definen 4 valores por cada una, como se muestran en la Tabla 3.18.

FM	***a***	***b***	***c***	***d***
Poca	0	0	20	35
Media	22.5	42.5	57.5	77.5
Mucha	65	80	100	100

Figura 3.18: Valores de pertenencia de la Humedad Relativa

Temperatura.

La Tabla 3.19 muestra los rangos utilizados en el SD para la temperatura, con base en el rango de la temperatura para la SN mostrado en la Tabla 3.15.

FM	Rango
Baja	13 a 24 °
Media	18 a 33 °
Alta	27 a 38 °

Figura 3.19: Rango de Temperatura

Para la variable de entrada temperatura, se utilizaron MF gaussianas, como se muestra en la Figura 3.20, puesto que se ajustan al rango de temperatura que favorece la proliferación de SN, mostrado en la Tabla 3.15. Las MF gaussianas tienen dos parámetros, uno de ellos (k) determina la curvatura y el otro (m) corresponde al punto central de la curva y se definen como:

$$MF = e^{-k(x-m)^2}$$

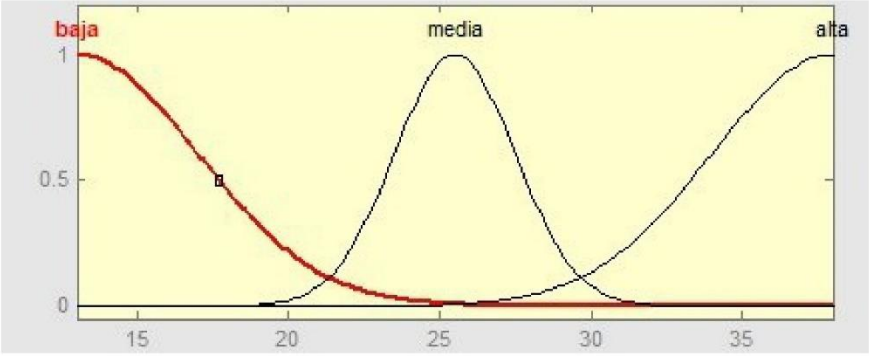


Figura 3.20: MF de la Temperatura

Reglas. Las reglas del SD se crearon basándose en las condiciones climáticas que propician la aparición de SN, presentadas en la Tabla 3.15. Los cambios hechos en la entrada afectan a la salida. Las combinaciones de las reglas establecidas se muestran en la Tabla 3.21.

Humedad/ Temperatura	Poca (0 – 35 %)	Media (22.5 - 77.5 %)	Mucha (65 – 100 %)
Baja (13 – 24 °)	Bajo	Bajo	Medio
Media (18 – 33 °)	Medio	Medio	Alto
Alta (27 – 38 °)	Bajo	Bajo	Medio

Figura 3.21: Reglas del Sistema Difuso

Discusión de Resultados.

Como resultado se obtuvo el modelado, simulación y validación del SD tipo TakagiSugeno, apartir de las variables climáticas de temperatura y humedad relativa, obteniendo el riesgo de proliferación de la SN. En la Figura 3.22 se observa el modelado del SD propuesto, con las dos variables climáticas de entrada (Temperatura y Humedad) y la variable de salida (Riesgo).

Figura 3.22: Modelo del SD para predecir la SN

Este sistema fue simulado con MATLAB, obteniendo la relación de las variables de entrada para determinar la variable de salida. Como se aprecia en la Figura 3.23, se puede observar que el mayor riesgo de proliferación respecto a la variable humedad se concentra en la parte alta, mientras que en la variable temperatura se concentra en la parte central.

Por otra parte, se puede apreciar que a menor humedad menor riesgo de proliferación, en cambio, el menor riesgo de proliferación en relación con la variable temperatura se ubica en los extremos inferior y superior.

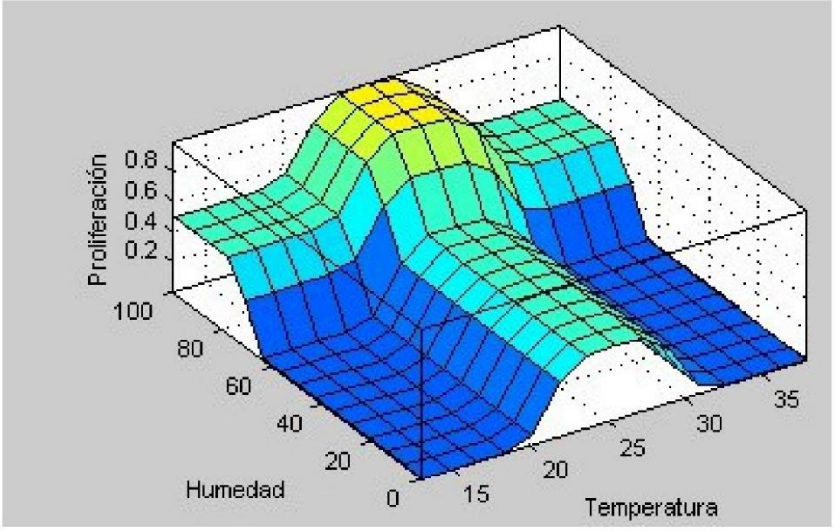


Figura 3.23: Simulación del SD

En la Figura 3.24 se puede observar el análisis que realiza MATLAB para determinar el riesgo, colocando el valor de las variables de entrada en sus respectivas FM, obteniendo la variable de salida.

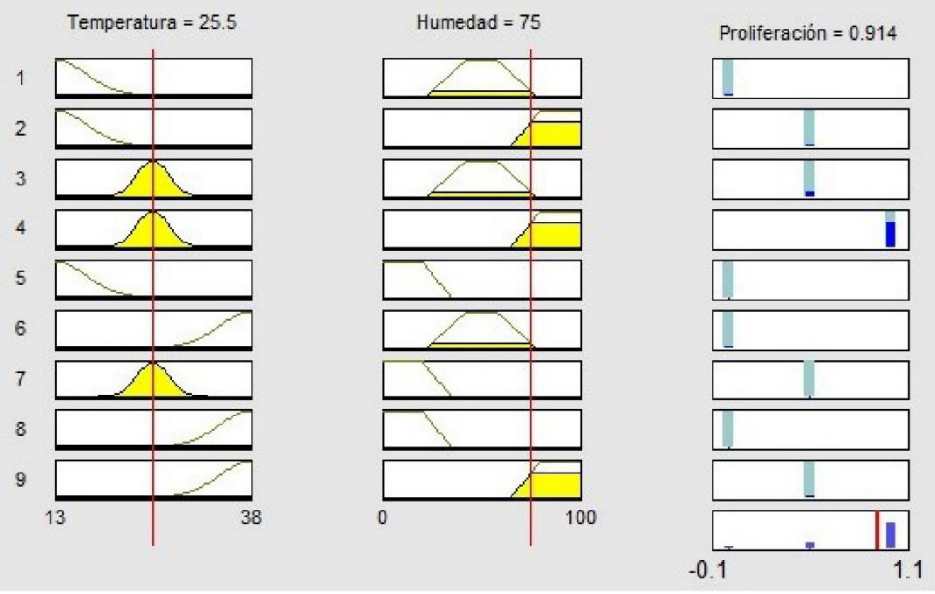


Figura 3.24: Resultados obtenidos del riesgo de proliferación en MATLAB.

Capítulo 4

Conclusiones

1. Los modelos difusos pueden ser una solución para multitud de problemas en diversos campos de aplicación, porque combinan la capacidad para repartir el conocimiento verbal en una representación matemática convencional (estructura del modelo), lo cual subsiguientemente puede ser puesto a punto usando datos de entrada y salida, que pueden ser entendidas por los usuarios de estos sistemas.

2. El modelo Takagi-Sugeno es una buena opción para sistemas de entrada-salida. Una ventaja es que reduce el número de reglas a evaluar y como característica distintiva con respecto a otros modelos, es que los consecuentes de las reglas difusas son una combinación lineal de los antecedentes.

3. En cuanto al desarrollo de aplicaciones, el modelado difuso es una forma de resolver problemas de los que no se tiene un modelo matemático; Cuando existe uno, no es conveniente emplear metodologías difusas.

4. Como propuestas a futuro, se desea profundizar en la aplicación de modelos difusos a un problema real en el pais, empleando los métodos propuestos en este trabajo.

Bibliografía

[1] Everitt B.S., et al. - Cluster analysis-Wiley (2011)

[2] János Abonyi, Balázs Feil - Cluster Analysis for Data Mining and System Identification-Birkhäuser (2007)

[3] James C. Bezdek, Pattern Recognition With Fuzzy Objective Function Algorithms,(1939)

[4] J.C. Bezdek and J.C. Dunn. Optimal fuzzy partitions: A heutristic for estimating the parameters in a mixture of normal distributions.(1975).

[5] L.A. Zadeh. Fuzzy Sets. Information and Control, 8:338–353, 1965.

[6] L.X. Wang and J.M. Mendel. Fuzzy Basis Functions, Universal Approximators,and Orthogonal Least-Squares Learning. IEEE Trans Neural Networks, 3(5):807–814, Sept. 1992.

[7] L.A. Zadeh. Fuzzy Sets. Information and Control, 8:338–353, 1965.

[8] W.Rudin, Principles of Mathematical Analysis. New York:McGraw-Hill, 1976.

[9] L.X. Wang. A course in Fuzzy Systems and Control. Prentice Hall, New York, USA, 1997.

[10] D. Driankov, H. Hellendoorn, and M. Reinfrank. An Introduction to Fuzzy Control. Springer-Verlag, Heidelberg, Germany, 1993.

[11] K.M. Passino and S. Yurkovic. Fuzzy Control. Addison-Wesley, New York, USA, 1998.

Anexos

Código del ejemplo en R del algoritmo K-means

```
library(datasets)
library(grid)
library(ggplot2) ## ploteo de los datos##
x <- read.csv(file="IRC.csv",head=TRUE,sep=",") ## Cargando la base de datos ##
y <- na.omit(x) ## omitir las lineas con valores faltantes##
y.scale <- as.data.frame(scale(y[,5,9]))
ycluster.scale <- kmeans(y.scale, 3)
plot(y$glucosa,y$creatinina,col = ycluster.scale$cluster,xlab = "glucosa",ylab = "creatinina")
aggregate(y[,5,9], by = list(ycluster.scale$cluster), mean)
ycluster.scale$cluster
## creando dos clusters ##
xcluster <- kmeans(y[,3:4], 2,nstart = 20)## creando dos clusters ##
names(xcluster) ## mostrando contenido del objeto ##
xcluster$cluster ## asignacion de las observaciones xcluster$totss ## inercia total
plot(y$glucosa,y$creatinina,col = xcluster$cluster,xlab = "glucosa",ylab = "creatinina")
aggregate(y[,5,9], by = list(ycluster.scale$cluster), mean)
## creando tres clusters ##
xcluster <- kmeans(y[,3:4], 3,nstart = 20) names(xcluster) ## mostrando contenido del objeto ##
xcluster$cluster ## asignacion de las observaciones xcluster$totss ## inercia total
plot(y$glucosa,y$creatinina,col = xcluster$cluster,xlab = "glucosa",ylab = "creatinina")
aggregate(y[,5,9], by = list(ycluster.scale$cluster), mean)
```

```
## creando cuatro clusters ##
xcluster < − kmeans(y[,3:4], 4,nstart = 20) names(xcluster) ## mostrando contenido
del objeto ##
xcluster$cluster ## asignacion de las observaciones xcluster$totss ## inercia total
plot(y$glucosa,y$creatinina,col = xcluster$cluster,xlab = “glucosa",ylab = “creatini-
na") aggregate(y[,5,9], by = list(ycluster.scale$cluster), mean)
# Encontrando la cantidad optima de cluster que se pueden formar #
sumbt < − kmeans(y[,3:4],1,nstart =20)$betweenss ## inicializando el vector
for (i in 2:10) sumbt[i] < − kmeans(y[,3:4],i,nstart =20)$betweenss
plot(1:10,sumbt, type = “b", xlab = “Num. clusters", ylab = “Suma de los cuadrados
inter grupos")
```

Código en R del ejemplo del algoritmo Gustafson-Kessel

```
data(iris)

iris_X < −iris
iris_X < − iris_X[,-5]

normalize < − function(M) {
#center data
means = apply(M,2,mean)
Xnorm = t(apply(M,1,function(x) {x-means}))
Xnorm}

encode < − function(M){ # put on hypershpere mins = apply(M,2,min)
maxs = apply(M,2,max)
ranges = maxs-mins
Xnorm = t(apply(M,1,function(x){ 2*(x-mins)/ranges-1}))
Xnorm = t(apply(Xnorm,1,function(x){x/norm(x,type="2")}))
Xnorm}

iris_norm = normalize(iris_X)
iris_norm = encode(iris_norm)

initCentroidsAndA < − function(k,n_features){
set.seed(123)
centroids = matrix(runif(k*n_features,-1,1), ncol = n_features)
centroids = normalize(centroids) centroids = encode(centroids)
A = vector(mode="list", length=k) for (i in seq(1,k)) {
A[[i]] = diag(x=1, ncol = n_features, nrow = n_features)}
list(ce = centroids, A = A)}
```

```
calculate_Mahalanobis <- function(X, cAndA) {
k = nrow(cAndA$ce)
centroids = cAndA$ce
A = cAndA$A
distances = c() for (i in seq(1,k)) {
dsq=as.matrix(apply(X,1,function(x){((t(x-centroids[i,])) %* %solve(A[[i]])) %* %(x-
centroids[i,])})) colnames(dsq) = paste(çentroid",i)
distances = cbind(distances,dsq)}
distances }
cAndA = initCentroidsAndA(3,n_features)
dst = calculate_Mahalanobis(iris_norm, cAndA)
calculate_memberships < − function(distances) {
mus = t(apply(distances, 1, function(x) { (1/x)/(sum(1/x)) }))
mus}
mships = calculate_memberships(dst)

update_centroids < − function(X,memberships) {
k = ncol(memberships)
n_features = ncol(X)
N = nrow(X) full = cbind(X,memberships)
centroids = c()
A = vector(mode="list", length=k)
for (i in seq(1,k)) {
den = sum((memberships[,i])2̂) num = 0
Fnum=matrix(data=0,nrow=n_features,ncol=n_features)
for (j in seq(1,N)) {
num = num + X[j,]*((memberships[j,i])2̂)}
cent = num/den centroids = rbind(centroids,cent)
```

```
for (j in seq(1,N)) {
Fnum=Fnum+(outer(((memberships[j,i])^2)*(X[j,]-centroids[i,]), X[j,]-centroids[i,]))}
Fi = Fnum/den
A[[i]] = ((det(Fi))^(1/n_features))*solve(Fi)}
list(ce = centroids, A = A)}
runGustafsonKessel <- function(X, k, epsilon) {
n_features = ncol(X)
centAndA = initCentroidsAndA(3, n_features)
tolerance = 2
iters = 0

while (tolerance > epsilon) {
dst = calculate_Mahalanobis(X, centAndA)
mships = calculate_memberships(dst)
newCentAndA = update_centroids(X,mships)
tolerance=((norm(centAndA$ce-newCentAndA$ce,type="2")))
#print(norm(newCentAndA$ce, type="2"))
#print(norm(centAndA$ce, type="2"))
#print("Tolerance")
#print(tolerance)
centAndA = newCentAndA
print(Çurrent centroids")
print(centAndA$ce)
iters = iters+1
#print(mships)}
list(dst = dst, centAndA = centAndA, mships = mships) }
z = runGustafsonKessel(iris_norm,3,0.05)
hard_partition <- function(memberships) {
apply(mships,1,which.max)}
```

```
library(rgl) library(car)
ind = sort(apply(iris_norm,2,var),index.return=TRUE, decreasing = TRUE)$ix[1:3]
labs = cnames[ind]
labels = hard_partition(z$mships)
sset = iris_norm[,ind]
x = sset[,1]
y = sset[,2]
z = sset[,3]
par3d("windowRect- c(0,0,400,400))
scatter3d(x = x, y = y, z = z, xlab=labs[1], ylab=labs[2], zlab = labs[3], labels = NULL,
groups = as.factor(labels),
surface = FALSE,
grid = FALSE,
ellipsoid=TRUE)
rgl.snapshot(filename = "plot_gk_iris.png")
```