



UNIVERSIDAD
DE GRANADA

DOBLE GRADO EN INGENIERÍA INFORMÁTICA Y MATEMÁTICAS

TRABAJO DE FIN DE GRADO

Diseño e implementación de mapas autoorganizativos para problemas complejos de agrupamiento con restricciones

Presentado por:

Sergio Cabezas González de Lara

Tutor:

Salvador García López

DECSAI

Mentor:

Germán González Almagro

DECSAI

Curso académico 2020-2021

Facultad de Ciencias
E.T.S. de Ingenierías Informática y de Telecomunicación



Diseño e implementación de mapas autoorganizativos para problemas complejos de agrupamiento con restricciones

Sergio Cabezas González de Lara

Sergio Cabezas González de Lara. *Diseño e implementación de mapas autoorganizativos para problemas complejos de agrupamiento con restricciones.*

Trabajo de fin de Grado. Curso académico 2020-2021.

**Responsable de
tutorización**

Salvador García López
DECSAI

Doble Grado en Ingeniería
Informática y Matemáticas

Mentor

Germán González Almagro
DECSAI

Facultad de Ciencias
E.T.S. de Ingenierías
Informática y de
Telecomunicación

Universidad de Granada

DECLARACIÓN DE ORIGINALIDAD

D. Sergio Cabezas González de Lara

Declaro explícitamente que el trabajo presentado como Trabajo de Fin de Grado (TFG), correspondiente al curso académico 2020-2021, es original, entendida esta, en el sentido de que no ha utilizado para la elaboración del trabajo fuentes sin citarlas debidamente.

En Granada a 9 de julio de 2021

Fdo: Sergio Cabezas González de Lara

Este trabajo va dedicado a mis abuelos Rafael y María Angustias, a quien junto a mis padres más ilusión les hace que termine este Doble Grado, más incluso que a mí. A mí tía Elena que junto a ellos me ha criado y ha estado siempre para cualquier cosa que necesitaba.

Pero muy especialmente, aparte de a ellos, este trabajo va dedicado a aquellos que me vieron hace 5 años empezar con miedo e ilusión este Doble Grado y que hoy, ya me están viendo desde el Cielo escribir estas líneas. A mis abuelos Francisco y Carmen, y a mis tías M^a Ángeles y M^a Cruz, ya no puedo celebrarlo con vosotros, pero sé que me seguís cuidando desde el Cielo.

Índice general

Índice de figuras	XIII
Índice de tablas	XV
Agradecimientos	XVII
Resumen	XIX
Summary	XXI
1. Introduction	XXI
2. Self-organizing maps for constrained clustering	XXII
3. Conclusion	XXII
3.1. Future Work	XXIII
Introducción	XXV
I. Preliminares	1
1. Definiciones y conceptos matemáticos.	3
1.1. Álgebra Lineal, Geometría y Topología	3
1.1.1. Espacios Vectoriales	3
1.1.2. Distancias	4
1.1.3. Topología	5
1.2. Probabilidad	6
1.2.1. Experimento Aleatorio	6
1.2.2. Espacio de probabilidad	6
1.2.3. Variable aleatoria	7
1.2.4. Probabilidad condicionada	9
1.2.5. Independencia de sucesos	9
1.3. Otros conceptos matemáticos	10
1.3.1. Funciones auxiliares	10
1.3.2. Análisis Vectorial	10
2. Definiciones y conceptos de computación	13
2.1. Complejidad Computacional	13
2.1.1. Introducción	13
2.1.2. Clases de complejidad	13
2.2. Inteligencia Artificial	14
2.2.1. Aprendizaje Automático	14

II. Mapas auto-organizativos (SOM) y Problema del agrupamiento con restricciones	17
3. El mapa auto-organizativo (SOM).	19
3.1. Cuantificación vectorial	19
3.1.1. Definición	19
3.1.2. Derivación del algoritmo de Cuantificación Vectorial.	20
3.2. El mapa auto-organizativo (SOM).	22
3.2.1. Definición de mapa auto-organizativo.	22
3.2.2. Formalización.	23
3.2.3. Entrenamiento del SOM	24
3.2.4. Visualización del proceso de adaptación.	26
3.2.5. Variante del SOM	29
3.3. Fundamentos Matemáticos del SOM	30
3.3.1. Proceso de Markov	30
3.3.2. Auto-organización	31
4. Problema del agrupamiento con restricciones	37
4.1. Descripción del problema	37
4.2. Estudio de la complejidad	38
4.3. Métodos desarrollados para el problema	39
5. Adaptación de los mapas autoorganizativos al clustering	41
5.1. Método de agrupación por contigüidad	41
III. Propuestas de algoritmos con mapas autoorganizativos para agrupamiento con restricciones	43
6. SCCSOM	45
6.1. Definición y explicación del SCCSOM	45
6.2. Explicación de métodos	45
7. CS2S	49
7.1. CS2GS	49
7.2. Adaptación a CS2S	50
7.3. Explicación de métodos	51
8. CopSoft-SOM	53
8.1. Cop-SOM	53
8.2. Adaptación a Cop-Soft SOM	55
8.3. Explicación de métodos	55
9. Ensembles	59
9.1. Un nuevo orden de asignación	59
9.1.1. La matriz de coasociación	59
9.1.2. El grado de certeza	60
9.2. Cambio de espacio de características	60

9.3. Algoritmos de Ensemble	61
9.3.1. SCCSOM-E	61
9.3.2. SCCSOM-EAC-SL	63
10. Implementación	65
10.1. Documentación	65
10.1.1. Funciones comunes	65
10.1.2. Métodos comunes a la clase SOM para clustering con restricciones	66
10.1.3. SCCSOM	68
10.1.4. CS2S	68
10.1.5. CopSoft-SOM	69
10.1.6. Funciones y métodos comunes de ensemble para SCCSOM	70
10.1.7. SCCSOM-E	70
10.1.8. SCCSOM-EAC-SL	71
IV. Experimentación	73
11. Configuración experimental	75
11.1. Generación de restricciones	76
11.2. Método de evaluación	77
11.2.1. Índice de Rand	77
11.2.2. Índice de Rand ajustado	78
11.3. Elección de los valores de los hiperparámetros	79
12. Resultados experimentales.	81
13. Conclusión y Trabajo Futuro	91
13.1. Conclusión	91
13.2. Trabajo Futuro	91
A. Resultados experimentales completos.	93
Bibliografía	113

Índice de figuras

1.1. Circunferencias unidad.	5
3.1. Mapa bidimensional de neuronas m_i con sus sublistas de instancias asociadas.	24
3.2. Dos ejemplos de vecindarios topológicos.	26
3.3. Adaptación del SOM 2D.	27
3.4. Adaptación del SOM 3D.	28
7.1. Esquema del algoritmo CS2GS.	50
8.1. Error de asignación de x_k en Cop-SOM.	53
8.2. Proceso de asignación de conflictos en Cop-SOM.	54
9.1. Modelo de ejecución del algoritmo Cop-SOM-E con soluciones al clustering por K-medias.	62
9.2. Esquema de ejecución del CC-EAC-SL.	64
10.1. Jerarquía de clases de SOMCC.	65
12.1. Gráfica comparativa SCCSOM.	88
12.2. Gráfica comparativa CS2S.	89
12.3. Gráfica comparativa CopSoft-SOM.	89
12.4. Gráfica comparativa SCCSOM-E.	90
12.5. Gráfica comparativa SCCSOM-EAC-SL.	90

Índice de tablas

4.1. Complejidad del problema.	38
9.1. Nuevo espacio de características.	61
11.1. Resumen de los conjuntos de datos utilizados para los experimentos.	75
11.2. Resumen de los conjuntos de restricciones utilizados para los experimentos.	76
11.3. Configuración de los parámetros para SCCSOM, CS2S y CopSoft-SOM.	79
11.4. Configuración de los parámetros para SCCSOM-E y SCCSOM-EAL-SL.	79
12.1. Resultados medios de las ejecuciones del SCCSOM.	83
12.2. Resultados medios de las ejecuciones del CS2S.	84
12.3. Resultados medios de las ejecuciones del CopSoft-SOM.	85
12.4. Resultados medios de las ejecuciones del SCCSOM-E.	86
12.5. Resultados medios de las ejecuciones del SCCSOM-EAL-SL.	87
A.1. Resultados de las ejecuciones del SCCSOM.	93
A.2. Resultados de las ejecuciones del CS2S.	97
A.3. Resultados de las ejecuciones del CopSoft-SOM.	101
A.4. Resultados de las ejecuciones del SCCSOM-E.	105
A.5. Resultados de las ejecuciones del SCCSOM-EAL-SL.	109

Agradecimientos

En primer lugar quiero agradecerles a mi tutor Salvador García y a mi mentor Germán Almagro su dedicación y esfuerzo; sus explicaciones han sido esenciales para ayudarme a desarrollar este proyecto.

Pero luego también quiero agradecerle a mis padres por todo lo que me han dado hasta llegar hasta aquí, por su cariño, por su comprensión y porque me han inculcado que con trabajo y esfuerzo todo se consigue; a mis hermanos que en estos 5 últimos años no he podido dedicarles toda la atención que quisiera y que desde luego se merecen. Y por último a mi novia, que me ha aguantado en estos dos últimos años tan difíciles de estudio, siempre mirando por mi bien.

Resumen

En resumen, este TFG va a consistir en el diseño de algoritmos basados en mapas auto-organizativos para la resolución del problema del clustering con restricciones y en la justificación matemática de los métodos que conllevan la resolución al problema.

En primer lugar para introducir al lector en el ámbito y tema del que se trata en este TFG, se introducen una serie de conceptos matemáticos que va a necesitar para la comprensión de los fundamentos matemáticos más avanzados en los que se basan los mapas auto-organizativos. Estos conceptos se podrían resumir en conceptos de espacios vectoriales, probabilidad, y análisis vectorial. Conceptos como derivadas, límites e integrales que son conocimientos de nivel intermedio en matemáticas, se consideran conocidos por el lector y por tanto no se van a explicar con detalle.

Además de estos conceptos matemáticos, se añadirá una pequeña introducción tanto a la complejidad de problemas como a la Inteligencia Artificial, centrándonos de esta última, en el Aprendizaje Automático y las Redes Neuronales, para luego poder explicar que es un mapa auto-organizativo de Kohonen (SOM).

Una vez vistos los conceptos básicos que nos ayuden a introducirnos en el tema, se expondrá qué es un mapa auto-organizativo de Kohonen y dos de los conceptos matemáticos sobre los que se fundamenta, la cuantificación vectorial y el fenómeno de auto-organización. Tras esto se definirá de forma matemática el problema de clustering con restricciones y se analizará su complejidad.

Tras esto, se explicará un método para adaptar la salida del SOM a un problema clásico de clustering; y se expondrá una propuesta de algoritmo para clustering con restricciones “soft” usando SOM. Junto a esta propuesta se añaden dos métodos más adaptados de la literatura de clustering con restricciones “hard” para SOM, y un método de “ensemble” usando SOM como modelo, al que se le aplican dos estrategias distintas de ordenación.

A continuación, se presenta el software implementado para realizar estos algoritmos, junto con una explicación de los métodos, clases y funciones simulando la clásica documentación que se ofrece en las librerías de código en Python.

Finalmente, se definen los conjuntos de datos, restricciones e hiperparámetros; se realiza la experimentación y se analizan los resultados de los experimentos sobre tablas y gráficas.

Summary

1. Introduction

Clustering is one of the most well-known and widely studied data analysis problems. It is a key area of research in the field of unsupervised learning, where there are no examples of how information should be treated.

Clustering has always been a powerful tool for knowledge discovery. Traditionally unsupervised, it has received renewed attention recently as it has been shown to produce better results when provided with new types of information, leading to a new type of semi-supervised learning: constrained clustering. This technique is a generalization of traditional clustering that takes into account additional information encoded by constraints. Constraints can be given in the form of instance-level binding and non-binding constraints, which is the focus of this work. It is then proposed to develop a semi-supervised machine learning method that includes this information to solve the clustering problem.

Semi-supervised learning (SSL) is a machine learning paradigm that arises from adding incomplete information to unsupervised learning. Following this paradigm we can incorporate background information to the clustering process, resulting in constrained clustering, which is the main topic of the study presented in this paper. The objective of constrained clustering is to find a partition of the data set that meets the characteristics of the results of a clustering method, in addition to satisfying a given set of constraints. It has been successfully applied in many fields of knowledge, including the following: advanced robotics applications, hyperspectral image classification, applied marketing, obstructive sleep apnea analysis, terrorist subcommunity detection, vocabulary maintenance policy for case-based reasoning systems, electoral district design, and lane search in GPS data among others.

A Kohonen self-organizing map, a neural network model designed for unsupervised learning and which is a largely unexplored method within semi-supervised learning, will be used as the basis for building the semi-supervised learning method for the problem.

The developed method receives the acronym SCCSOM and in this work we will perform an experimental study of it, comparing it with other existing algorithms in the literature that can be adapted to the problem.

Finally, this work also intends to make a detailed explanation of the whole mathematical formalism on which the problem and the self-organizing maps are based, and therefore, on the formalism on which the proposed algorithm is based. One objective is to make this formalism understandable for a person with average knowledge in Mathematics (derivatives, integrals, limits, ...).

2. Self-organizing maps for constrained clustering

In summary, this dissertation will consist of the design of algorithms based on self-organizing maps for the resolution of the clustering problem with constraints and the mathematical justification of the methods that lead to the resolution of the problem.

First of all, in order to introduce the reader to the scope and subject matter of this dissertation, a series of mathematical concepts are introduced that will be necessary for the understanding of the more advanced mathematical foundations on which self-organizing maps are based. These concepts could be summarized as concepts of vector spaces, probability, and vector analysis. Concepts such as derivatives, limits, and integrals, which are intermediate level knowledge in mathematics, are considered to be known to the reader and therefore will not be explained in detail.

In addition to these mathematical concepts, a short introduction to both problem complexity and Artificial Intelligence will be added, focusing on Machine Learning and Neural Networks, and then explaining what a Kohonen Self-Organizing Map (SOM) is.

Once we have seen the basic concepts that help us to introduce the subject, we will explain what a Kohonen self-organizing map is and two of the mathematical concepts on which it is based, vector quantization and the phenomenon of self-organization. After this, the clustering problem with constraints will be defined mathematically and its complexity will be analyzed.

After this, a method to adapt the SOM output to a classical clustering problem will be explained; and a proposed algorithm for clustering with “soft” constraints using SOM will be presented. Along with this proposal, two more methods adapted from the “hard” constrained clustering literature for SOM, and an “ensemble” method using SOM as a model, to which two different sorting strategies are applied, are added.

The software implemented to perform these algorithms is presented below, along with an explanation of the methods, classes and functions simulating the classic documentation provided in Python code libraries.

Finally, the data sets, constraints and hyperparameters are defined; the experimentation is performed and the results of the experiments are analyzed on tables and graphs.

3. Conclusion

In this paper we have proposed the SCCSOM algorithm, a “soft” constrained clustering method using a self-organizing map. As already mentioned in the introduction, the approach of using “soft” constraints had not been treated for this problem using self-organizing maps, there were only proposals with “hard” constraints. Therefore, in addition to our proposal, we have had to adapt the existing ones to the case of “soft” constraints, such as SC2GS and Cop-SOM.

After experimentation with the three algorithms, the suitability of my proposal has been

demonstrated, since for the data sets tested, SCCSOM improves by far over the other two algorithms as the number of constraints increases. This has come to demonstrate both the suitability of the SCCSOM algorithm for the problem, since it presents very good results, and its usefulness since it is an approach given to the constraint clustering problem that had never been done with self-organizing maps and because the adaptation of algorithms that had already been implemented for a similar problem are not able to achieve good results.

In addition, two ensemble methods, SCCSOM-E and SCCSOM-EAC-SL, which are based on the proposed SCCSOM algorithm for clustering with “soft” constraints, have been implemented. After experimentation, it has been shown that self-organizing maps offer a very good result when used as the last layer of these multiple learning systems, opening the possibility of implementing self-organizing maps with several layers of depth or more complex systems with a final layer SOM.

It should also be noted that the objective of deepening the mathematical basis has been achieved, explaining in detail the Markov processes and vector quantization, as well as demonstrating the phenomenon of self-organization. Together with this, it has been possible to give to a current and classic problem of computer science an accessible and understandable mathematical formalism for any person with intermediate knowledge in Mathematics.

Therefore we can conclude that the work has achieved the objectives sought, demonstrating the suitability of the use of self-organizing constraints and maps for the clustering problem.

3.1. Future Work

In order to continue developing the topic addressed in the future, two lines of research are proposed, one more mathematical and the other more of development and implementation of algorithms.

The more mathematical part could consist of applying more sophisticated statistical techniques, such as statistical tests or Bayesian models, to the experimental results in order to draw more conclusions than those drawn in this work. Another possible line of mathematical research could be the design of more complex topologies for the maps or the design of distances in the map where the weights of the neurons, the epochs and above all the constraints, and not only the topology of the map, have an influence.

A line of research more focused on the development and implementation of algorithms could focus on the study of different variants of the SOM, such as the GSOM, and following the strategy that has been done in this work for the basic SOM, adapt the problem of clustering with constraints to these new types of self-organizing maps. Another possible line of research could be to adapt the SOM library for constrained clustering so that its more cumbersome methods such as network training could be executed in parallel with different threads, a function that is not yet available. In addition, a hybridization of the self-organizing map with a genetic algorithm could be investigated to help further improve the search for the optimal cluster configuration.

Summary

Finally, extending the last part of the previous section, an interesting line of research would be to investigate the possibility of creating multilayer self-organizing maps and the possible applications of these self-organizing maps that would form multilayer networks of neurons, resembling existing dense networks.

Introducción

El agrupamiento o clustering es uno de los problemas de análisis de datos más conocidos y ampliamente estudiados. Constituye un área de investigación clave en el campo del aprendizaje no supervisado, en el que no hay ejemplos sobre cómo debe tratarse la información.

El clustering siempre ha sido una poderosa herramienta para el descubrimiento de conocimiento. Tradicionalmente no supervisada, ha recibido una atención renovada recientemente al demostrar que produce mejores resultados cuando se le proporcionan nuevos tipos de información, lo que ha dado lugar a un nuevo tipo de aprendizaje semisupervisado: la agrupación con restricciones. Esta técnica es una generalización de la agrupación tradicional que tiene en cuenta información adicional codificada por restricciones. Las restricciones pueden ser dadas en forma de restricciones de enlace a nivel de instancia y de no enlace, que es el foco de este trabajo. Entonces se propone desarrollar un método de aprendizaje automático semisupervisado que incluya esta información para resolver el problema de clustering.

El aprendizaje semisupervisado (SSL) es un paradigma de aprendizaje automático que surge de añadir información incompleta al aprendizaje no supervisado. Siguiendo este paradigma podemos incorporar información de fondo al proceso de clustering, resultando en un clustering con restricciones, que es el tema principal del estudio presentado en este trabajo. El objetivo del clustering con restricciones es encontrar una partición del conjunto de datos que cumpla con las características propias de los resultados de un método de clustering, además de satisfacer un determinado conjunto de restricciones. Se ha aplicado con éxito en muchos campos del conocimiento, entre los que cabe destacar los siguientes: aplicaciones de robótica avanzada, clasificación de imágenes hiperespectrales, marketing aplicado, análisis de apnea obstructiva del sueño, detección de subcomunidades terroristas, política de mantenimiento de vocabulario para sistemas de razonamiento basados en casos, diseño de distritos electorales, y búsqueda de carriles en datos GPS entre otros.

Como base para construir el método de aprendizaje semisupervisado para el problema se usará un mapa auto-organizativo de Kohonen, un modelo de red neuronal diseñado para aprendizaje no supervisado y que es un método muy inexplorado dentro del aprendizaje semisupervisado.

El método desarrollado recibe el acrónimo de SCCSOM y en este trabajo efectuaremos un estudio experimental de él, comparándolo con los demás algoritmos existentes en la literatura que se puedan adaptar al problema.

Finalmente, este trabajo también tiene la intención de hacer una explicación detallada de todo el formalismo matemático sobre el que se apoyan el problema y los mapas auto-organizativos y por tanto, en el formalismo en el que se apoya el algoritmo propuesto. Un objetivo es conseguir hacer entendible este formalismo para una persona con conocimientos medios en Matemáticas (derivadas, integrales, límites, ...).

Parte I.

Preliminares

1. Definiciones y conceptos matemáticos.

1.1. Álgebra Lineal, Geometría y Topología

Como vamos a tratar con muestras de valores de señales relacionadas espacial y temporalmente que constituyen patrones, necesitamos un marco matemático para la descripción de sus interrelaciones cuantitativas. Este marco suele proporcionarlo el formalismo vectorial generalizado, por lo que introduciremos conceptos de espacios vectoriales que ayudarán al manejo del formalismo matemático del mapa autoorganizado. Por otra parte, las operaciones en los espacios vectoriales pueden manipularse cómodamente mediante operaciones algebraicas.

1.1.1. Espacios Vectoriales

Definición 1.1. Un **espacio vectorial real** es una terna $(V, +, \cdot)$, donde V es un conjunto no vacío y, $+$ y \cdot son dos operaciones del tipo:

$$+ : V \times V \rightarrow V \quad \text{y} \quad \cdot : \mathbb{R} \times V \rightarrow V,$$

que se denominan “suma de vectores” y “producto por escalares” respectivamente, con las siguientes propiedades:

1. $u + (v + w) = (u + v) + w \quad \forall u, v, w \in V$.
2. $u + v = v + u \quad \forall u, v \in V$.
3. $\exists e \in V$ tal que $e + v = v + e = v \quad \forall v \in V$.
4. $\forall v \in V \exists w$ tal que $v + w = w + v = e$.
5. $\lambda(\mu v) = (\lambda\mu)v, \quad \forall v \in V, \forall \lambda, \mu \in \mathbb{R}$.
6. $\lambda(u + v) = \lambda u + \lambda v$ y $(\lambda + \mu)v = \lambda v + \mu v, \quad \forall u, v \in V$ y $\forall \lambda, \mu \in \mathbb{R}$.
7. $1v = v, \quad \forall v \in V$.

De forma abreviada, diremos que V es un espacio vectorial real. A los elementos de V los llamamos vectores y a los de \mathbb{R} , escalares.

Definición 1.2. Sea V un espacio vectorial real, se define el **producto escalar** asociado a V como la aplicación:

$$(\cdot | \cdot) : V \times V \rightarrow \mathbb{R}$$

que verifica las propiedades:

1. $(u|v) = (v|u) \quad \forall u, v \in V$.
2. $(u + v|w) = (u|w) + (v|w) \quad \forall u, v, w \in V$.

1. Definiciones y conceptos matemáticos.

$$3. (\lambda u|v) = \lambda(u|v) \quad \forall u, v \in V, \forall \lambda \in \mathbb{R}.$$

Al par $(V, (\cdot|\cdot))$ se le denomina **espacio vectorial euclídeo**.

Definición 1.3. Un ejemplo de espacio vectorial euclídeo es el espacio vectorial $V = \mathbb{R}^n$, al que se le asocia el denominado **producto escalar usual**. Dados dos vectores $x, y \in \mathbb{R}^n$, se define el producto escalar usual como:

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

Definición 1.4. Un espacio vectorial V se dice que es un **espacio normado** si en él se puede definir una norma, es decir, una aplicación

$$\|\cdot\| : V \rightarrow \mathbb{R}$$

que verifica:

1. $\forall x \in V, \|x\| \geq 0$ y $\|x\| = 0 \Leftrightarrow x = 0$.
2. $\forall x \in V$ y $\forall k \in \mathbb{R}, \|kx\| = |k| \cdot \|x\|$ donde $|\cdot|$ es el valor absoluto.
3. $\forall x, y \in V, \|x + y\| \leq \|x\| + \|y\|$.

Generalmente se denota como $(V, \|\cdot\|)$ al espacio vectorial normado. En todo espacio vectorial normado se puede definir una función distancia de la forma

$$d : V \times V \rightarrow \mathbb{R}$$

con

$$d(x, y) = \|x - y\|;$$

y con la cual (V, d) es un **espacio métrico**.

Definición 1.5. Se define la **norma euclídea** de un vector $x \in \mathbb{R}^n$ como la raíz cuadrada del producto escalar usual de x por sí mismo, es decir como:

$$\begin{array}{ccc} \|\cdot\| : \mathbb{R}^n & \rightarrow & \mathbb{R} \\ x & \rightarrow & \|x\| \end{array}$$

con

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

1.1.2. Distancias

Para la comparación entre vectores (neuronas y entradas), hay que introducir medidas que describan la distancia entre sus representaciones o la similitud de las mismas. Se necesita para poder establecer un vecindario entre las neuronas del mapa.

Definición 1.6. Sean $x, y \in \mathbb{R}^n$, se define la distancia euclídea entre x e y como:

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

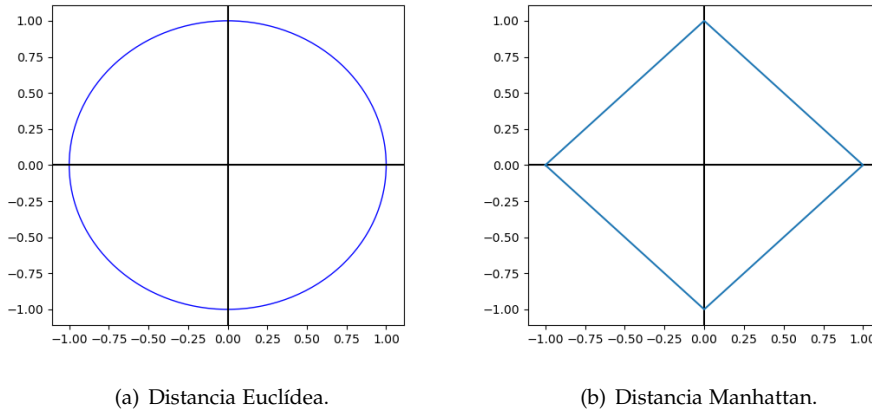


Figura 1.1.: Circunferencias unidad.

Definición 1.7. Sean $x, y \in \mathbb{R}^n$, se define la distancia de Manhattan entre x e y como:

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

1.1.3. Topología

Con tal de entender las proposiciones matemáticas que se expondrán en el desarrollo del TFG, se necesitará conocer un mínimo sobre conceptos topológicos que existen en la Teoría de Conjuntos. Por tanto, se van a introducir conceptos como el de interior y frontera de un conjunto que ayudarán a manejar conceptos también topológicos como el de recubrimiento y el de teselación que se verán en la [Def. 3.3](#).

Definición 1.8. Sea (X, d) un espacio métrico, $a \in X$ y $r \in \mathbb{R}^+$. Se denomina **bola** (abierta) centrada en a y de radio r al siguiente subconjunto de X :

$$B_r(a) = \{x \in X : d(x, a) < r\}.$$

Definición 1.9. Sea (X, d) un espacio métrico y $A \subset X$. Se define el **interior** de A como:

$$\text{int}(A) = \{a \in A : \exists r > 0, B_r(a) \subset A\}.$$

En este caso, un punto $a \in A$ es parte del interior de A si y sólo si existe una bola abierta de centro a y radio $r > 0$ contenida en A .

Definición 1.10. Sea (X, d) un espacio métrico y $A \subset X$. Se define la **frontera** de A como:

$$\text{fr}(A) = \{x \in X : \forall r > 0, B_r(x) \cap A \neq \emptyset \neq B_r(x) \cap (X - A)\}.$$

1.2. Probabilidad

La identificación y clasificación de las observaciones naturales se basa en propiedades estadísticas y en distribuciones de probabilidad. Por lo tanto, para nuestro mapa autoorganizado vamos a necesitar manejar conceptos básicos ya que nuestro mapa va a representar una proyección de la distribución de probabilidad del conjunto de entrada. Esta proyección vendrá determinada según las ocurrencias de las instancias.

Concretamente vamos a definir qué es un experimento aleatorio, y cómo a partir de ahí se construye la Teoría de Probabilidad, llegando a exponer conceptos necesarios para la explicación de la fundamentación matemática del SOM, que se verá más adelante, como es la independencia de sucesos y la probabilidad condicionada.

1.2.1. Experimento Aleatorio

Podemos definir un experimento aleatorio como aquel que satisface las siguientes condiciones:

- Todos sus posibles resultados son conocidos por anticipado.
- Puede repetirse sucesivamente en las mismas condiciones.
- Bajo las mismas condiciones, puede dar lugar a distintos resultados.
- No se puede prever su resultado en una experiencia particular.

1.2.2. Espacio de probabilidad

Definición 1.11. Se denomina **suceso** a cualquier característica, hecho o proposición lógica que pueda formularse en relación a un experimento aleatorio, cuya ocurrencia o no ocurrencia pueda ser observada tras la realización del experimento.

Destacan cuatro tipos de sucesos:

- **Suceso elemental.** Suceso formado por un único elemento del espacio muestral. Corresponde a cada uno de los resultados posibles del experimento aleatorio.
- **Suceso compuesto.** Suceso que consta de dos o más sucesos elementales.
- **Suceso seguro.** Suceso que ocurre siempre. Consta de todos los sucesos elementales del espacio muestral y se identifica con el espacio muestral total Ω .
- **Suceso imposible.** Suceso que no ocurre nunca. No contiene ningún elemento del espacio muestral y se identifica con \emptyset .

Definición 1.12. Se define como **espacio muestral** al conjunto formado por todos los sucesos elementales asociados a un experimento aleatorio. Se le designa por Ω .

Definición 1.13. Sea $\mathcal{A} \subset \mathcal{P}(\Omega)$ una clase de sucesos no vacía. Decimos que \mathcal{A} tiene estructura de σ -álgebra si se verifica que es cerrada para complementarios y uniones numerables. Esto se produce si se verifican las condiciones:

1. $\forall A \in \mathcal{A}$ se verifica que su complementario $\bar{A} \in \mathcal{A}$.
2. Dada una colección numerable de sucesos, $\{A_i\}_{i \in \mathbb{N}} \in \mathcal{A}$, se verifica que

$$A_1 \cup A_2 \cup A_3 \cup \dots = \bigcup_{i=1}^{\infty} A_i \in \mathcal{A}.$$

Al par formado por un espacio muestral Ω y una clase de conjuntos \mathcal{A} con estructura de σ -álgebra, esto es (Ω, \mathcal{A}) , se le denomina **espacio medible** y a los conjuntos de \mathcal{A} , conjuntos medibles.

Definición 1.14. Sobre el conjunto de números reales \mathbb{R} , se define la **σ -álgebra de Borel** como la mínima clase de conjuntos con estructura de σ -álgebra que contiene a todos los intervalos de \mathbb{R} . Esto es, si \mathcal{Y} denota la clase de intervalos de \mathbb{R} , la σ -álgebra de Borel, \mathcal{B} , es una clase de conjuntos de \mathbb{R} , $(\mathcal{B} \subset \mathcal{P}(\mathbb{R}))$, tal que:

1. $\mathcal{B} \supset \mathcal{Y}$.
2. \mathcal{B} es σ -álgebra.
3. Si $\mathcal{A} \subset \mathcal{P}(\mathbb{R})$ es una σ -álgebra tal que $\mathcal{A} \supset \mathcal{Y}$, entonces $\mathcal{A} \supset \mathcal{B}$.

Al par $(\mathbb{R}, \mathcal{B})$ se le denomina **Espacio de Borel**. Los elementos de \mathcal{B} se denominan Conjuntos de Borel o Borelianos.

Definición 1.15. Dado un espacio muestral Ω asociado a un determinado experimento aleatorio y una clase de conjuntos de Ω con estructura de σ -álgebra, \mathcal{A} , se define una **función de probabilidad** o medida de probabilidad como una función de conjunto P tal que

$$P : \mathcal{A} \rightarrow [0, 1]$$

y que verifica los siguientes axiomas:

1. $P(A) \geq 0, \forall A \in \mathcal{A}$.
2. $P(\Omega) = 1$.
3. Sea $\{A_i\}_{i \in \mathbb{N}} \subset \mathcal{A}$ con $A_i \cap A_j = \emptyset \forall i \neq j$, entonces

$$P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i).$$

Así, $P(A) \forall A \in \mathcal{A}$ denota la probabilidad del suceso A . A la terna formada por el espacio muestral Ω , la σ -álgebra \mathcal{A} y la función de probabilidad P , (Ω, \mathcal{A}, P) , se le denomina espacio probabilístico o **espacio de probabilidad**.

1.2.3. Variable aleatoria

En esta sección se va a explicar lo que es una variable aleatoria y según los tipos, discreto o continuo, se abordará la explicación de su función de probabilidad y el cálculo de su esperanza.

1. Definiciones y conceptos matemáticos.

Definición 1.16. Se define una **variable aleatoria**, X , como una función medible sobre un espacio de probabilidad, es decir una función

$$X : (\Omega, \mathcal{A}, P) \rightarrow (\mathbb{R}, \mathcal{B})$$

tal que la imagen inversa de cualquier conjunto de Borel es medible (un suceso del σ -álgebra \mathcal{A}); esto es,

$$X^{-1}(B) \in \mathcal{A} \quad \forall B \in \mathcal{B}$$

Definición 1.17. Una variable aleatoria induce una medida de probabilidad P_X sobre el espacio de Borel $(\mathbb{R}, \mathcal{B})$ dando lugar a un espacio de probabilidad $(\mathbb{R}, \mathcal{B}, P_X)$ de la siguiente forma:

$$P_X(B) = P[X^{-1}(B)] \quad \forall B \in \mathcal{B}.$$

Se define la **función de distribución** de una variable aleatoria X como una función

$$F_X : \mathbb{R} \rightarrow [0, 1]$$

de la siguiente forma

$$F_X(x) = P_X((-\infty, x]) \quad \forall x \in \mathbb{R}.$$

Definición 1.18. Una **variable aleatoria** X se dice que es de tipo discreto o simplemente **discreta** si existe un conjunto numerable $E \subset \mathbb{R}$ tal que $P[X \in E] = 1$; es decir, si toma a lo sumo una cantidad numerable de valores posibles.

Sea X una variable aleatoria discreta con valores en $E = \{x_i, i = 1, 2, \dots\}$. A la función $P : E \rightarrow [0, 1]$ que a cada valor de E le asigna la probabilidad con que la variable toma ese valor se le denomina **función masa de probabilidad** de la variable aleatoria y se tiene que

$$x_i \rightarrow P[X = x_i] = p_i \quad i = 1, 2, \dots$$

y verifica $0 \leq P[X = x_i] \leq 1, i = 1, 2, \dots$ y $\sum_i P[X = x_i] = 1$.

Definición 1.19. Una **variable aleatoria** X es de tipo continuo o simplemente **continua** si su función de distribución es absolutamente continua, es decir, si existe una función $f : \mathbb{R} \rightarrow \mathbb{R}$ tal que

$$F(x) = \int_{-\infty}^x f(t) dt \quad \forall x \in \mathbb{R}.$$

A dicha función se le llama **función de densidad** de la variable aleatoria y verifica las siguientes condiciones:

1. $f(x) \geq 0 \quad \forall x \in \mathbb{R}$.
2. f es integrable Riemann ya que F tiene a lo sumo un número finito de discontinuidades sobre cada intervalo finito de \mathbb{R} .
3. $\int_{-\infty}^{\infty} f(x) dx = 1$.

Definición 1.20. Sea $X : (\Omega, \mathcal{A}, P) \rightarrow (\mathbb{R}, \mathcal{B}, P_X)$ una **variable aleatoria discreta** con valores en $M \subset \mathbb{R}$. Se define la **esperanza o valor esperado de X** como

$$E[X] = \sum_{x \in M} x P[X = x]$$

siempre que dicha suma sea absolutamente convergente, es decir si

$$\sum_{x \in M} |x|P[X = x] < \infty.$$

En caso de no convergencia absoluta de la serie se dice que no existe la esperanza de dicha variable aleatoria.

Definición 1.21. Sea $X : (\Omega, \mathcal{A}, P) \rightarrow (\mathbb{R}, \mathcal{B}, P_X)$ una **variable aleatoria continua** con función de densidad f_X . Se define la **esperanza o valor esperado de X** como

$$E[X] = \int_{-\infty}^{\infty} x f_X(x) dx$$

siempre que dicha integral sea absolutamente convergente, es decir si

$$\int_{-\infty}^{\infty} |x| f_X(x) dx < \infty.$$

En caso de no convergencia absoluta de la serie se dice que no existe la esperanza de dicha variable aleatoria.

Definición 1.22. Decimos que una sucesión de variables aleatorias $\{X_n\}_{n \in \mathbb{N}}$ **converge de forma casi segura** (o con probabilidad uno) a una variable aleatoria límite X , cuando el conjunto de sucesos ω , que cumple que $X(\omega)$ es el límite de la sucesión $\{X_n(\omega)\}$, tiene probabilidad uno.

1.2.4. Probabilidad condicionada

Definición 1.23. Sea (Ω, \mathcal{A}, P) un espacio de probabilidad y A un suceso ($A \in \mathcal{A}$) tal que $P(A) > 0$. Para cualquier otro suceso $B \in \mathcal{A}$, se define la **probabilidad condicionada** de B dado A como

$$P(B/A) = \frac{P(B \cap A)}{P(A)}.$$

Definición 1.24. Sean X e Y variables aleatorias sobre el mismo espacio de probabilidad. Se define la **esperanza condicionada** de X a Y como la variable aleatoria, que se denota $E[X/Y]$, que toma el valor $E[X/Y = y]$ cuando $Y = y$, siendo

$$E[X/Y = y] = \begin{cases} \sum x P[X = x/Y = y] & \text{si } (X, Y) \text{ es de tipo discreto} \\ \int_{\mathbb{R}} x f(x/y) dx & \text{si } (X, Y) \text{ es de tipo continuo y } f_Y(y) \neq 0 \end{cases}$$

y suponiendo que tales sumas o integrales son absolutamente convergentes.

1.2.5. Independencia de sucesos

Definición 1.25. Sea (Ω, \mathcal{A}, P) un espacio de probabilidad y $A \in \mathcal{A}$ con $P(A) > 0$. La ocurrencia del suceso A puede alterar la probabilidad de ocurrencia de cualquier otro suceso $B \in \mathcal{A}$. Al estudiar dichas probabilidades, surge el concepto de **independencia de sucesos** ya que pueden darse los siguientes casos:

1. Definiciones y conceptos matemáticos.

- Si $P(B/A) \neq P(B)$, es decir la ocurrencia del suceso A modifica la probabilidad de ocurrencia de B . Diremos entonces que el suceso B depende del suceso A .
- Si $P(B/A) = P(B)$, es decir, la ocurrencia del suceso A no tiene ningún efecto sobre el suceso B , se dice que el suceso B es independiente del suceso A .

1.3. Otros conceptos matemáticos

Además de los conceptos descritos en las dos secciones anteriores, se van a necesitar conceptos de análisis como es el gradiente y una serie de funciones que van a ser necesarias en el desarrollo del entrenamiento del mapa autoorganizado.

1.3.1. Funciones auxiliares

Definición 1.26. Se define la **delta de Kronecker** como una función de dos variables que toma el valor 1 si las dos variables son iguales y 0 en caso contrario. Esta función se define como:

$$\delta_{ij} : \mathbb{Z} \times \mathbb{Z} \rightarrow \{0, 1\},$$

de forma que

$$\delta_{ij} = \begin{cases} 0 & \text{si } i \neq j \\ 1 & \text{si } i = j \end{cases}.$$

Definición 1.27. Sea X un conjunto y $A \subset X$ un subconjunto. Se define la **función característica** o indicadora del subconjunto $A \subset X$ como una función

$$\chi_A : X \rightarrow \{0, 1\}$$

definida como

$$\chi_A(x) = \begin{cases} 1 & \text{si } x \in A \\ 0 & \text{si } x \notin A \end{cases}.$$

La función χ_A en ocasiones se expresa como 1_A o I_A .

Definición 1.28. Sea $f : X \rightarrow \mathbb{R}^n$ una función real cuyo dominio es un conjunto arbitrario X , entonces el **sopORTE** de f , denotado por $\text{supp}(f)$, es el conjunto de puntos en X donde f no es cero, es decir

$$\text{supp}(f) = \{x \in X | f(x) \neq 0\}.$$

Definición 1.29. Se define la **función gaussiana** o campana de Gauss como la función definida por la expresión:

$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}},$$

donde a , b y c son constantes reales con $(c > -1)$.

1.3.2. Análisis Vectorial

Definición 1.30. Un **campo escalar** en una región $U \subseteq \mathbb{R}^n$ es una función

$$f : U \rightarrow \mathbb{R}$$

en la que a cada punto $x \in U$, se le asigna un número o escalar $f(x)$.

Definición 1.31. Un **campo vectorial** sobre un subconjunto $X \subseteq \mathbb{R}^n$ es una función con valores vectoriales:

$$F : X \rightarrow \mathbb{R}^m.$$

Definición 1.32. Sea $f : \mathbb{R}^n \rightarrow \mathbb{R}$ un campo escalar entonces el gradiente de f en x se define como el campo vectorial

$$\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}^n$$

cuyas componentes son las derivadas parciales del campo escalar, esto es:

$$\nabla f(\mathbf{r}) = \left(\frac{\partial f(\mathbf{r})}{\partial x_1}, \dots, \frac{\partial f(\mathbf{r})}{\partial x_n} \right).$$

2. Definiciones y conceptos de computación

2.1. Complejidad Computacional

La **teoría de la complejidad computacional** es una rama de la teoría de la computación que se centra en la clasificación de los problemas computacionales de acuerdo con su dificultad inherente y en la relación entre dichas clases de complejidad.

2.1.1. Introducción

Un problema se cataloga como “inherentemente difícil” si su solución requiere de una cantidad significativa de recursos computacionales, sin importar el algoritmo utilizado. La teoría de la complejidad computacional formaliza dicha aseveración, introduciendo modelos de computación matemáticos para el estudio de estos problemas y la cuantificación de la cantidad de recursos necesarios para resolverlos, como tiempo y memoria.

Una de las metas de la teoría de la complejidad computacional es determinar los límites prácticos de qué es lo que se puede hacer en un computador y qué no. Otros campos relacionados con la teoría de la complejidad computacional son el análisis de algoritmos y la teoría de la computabilidad. Una diferencia significativa entre el análisis de algoritmos y la teoría de la complejidad computacional, es que el primero se dedica a determinar la cantidad de recursos requeridos por un algoritmo en particular para resolver un problema, mientras que la segunda, analiza todos los posibles algoritmos que pudieran ser usados para resolver el mismo problema.

2.1.2. Clases de complejidad

La teoría de la complejidad computacional trata de clasificar los problemas que pueden, o no pueden ser resueltos con una cantidad determinada de recursos. Estos problemas son clasificados en diferentes **clases de complejidad**, pero sólo vamos a centrarnos en dos:

- **Clase P.** Clase de complejidad de los problemas de decisión que pueden ser resueltos en tiempo polinómico calculado a partir de la entrada por una máquina de Turing determinista.
- **Clase NP.** Clase de complejidad de los problemas de decisión que pueden ser resueltos en tiempo polinómico calculado a partir de la entrada por una máquina de Turing no determinista.
 - **Clase NP-completo.** Subconjunto de los problemas de decisión en NP tal que todo problema en NP se puede reducir en cada uno de los problemas de NP-completo. Se puede decir que los problemas de NP-completo son los problemas más difíciles de NP y muy probablemente no formen parte de la clase de complejidad P.

2.2. Inteligencia Artificial

La **Inteligencia Artificial (IA)** es el campo científico de la informática que se centra en la creación de programas y mecanismos con la finalidad de lograr que los computadores realicen tareas que, por el momento, los humanos hacen mejor. (Rich & Knight, 1991).

En ciencias de la computación, una máquina “inteligente” ideal es un agente flexible que percibe su entorno y lleva a cabo acciones que maximicen sus posibilidades de éxito en algún objetivo o tarea. Además, en su uso coloquial el término inteligencia artificial se aplica cuando una máquina imita las funciones “cognitivas” que los humanos asocian con otras mentes humanas.

Dentro de los problemas que resuelve la inteligencia artificial se encuentran los sistemas expertos, el aprendizaje automático, la visión por computador y el manejo y control de robots, entre otros muchos. En este TFG nos centraremos en el aprendizaje automático.

2.2.1. Aprendizaje Automático

EL aprendizaje automático es el subcampo de las ciencias de la computación y una rama de la inteligencia artificial, cuyo objetivo es desarrollar técnicas que permitan que las computadoras aprendan. Se dice que un agente aprende cuando su desempeño mejora con la experiencia. Por tanto, se podría decir que el aprendizaje automático es el estudio de algoritmos de computación que mejoran automáticamente su rendimiento gracias a la experiencia. De forma más concreta, los investigadores del aprendizaje automático buscan algoritmos y heurísticas para obtener soluciones complejas a partir de grandes muestras de datos.

En muchas ocasiones el campo de actuación del aprendizaje automático se solapa con el de la estadística inferencial, ya que las dos disciplinas se basan en el análisis de datos. Sin embargo, el aprendizaje automático incorpora las preocupaciones de la complejidad computacional de los problemas. Muchos problemas son de clase NP-completo ([Subsección 2.1.2](#)), por lo que gran parte de la investigación realizada en aprendizaje automático está enfocada al diseño de soluciones factibles a esos problemas. El aprendizaje automático también está estrechamente relacionado con el agrupamiento o clustering y el reconocimiento de patrones. Además también puede ser visto como un intento de automatizar algunas partes del método científico mediante métodos matemáticos. Por lo tanto es un proceso de inducción del conocimiento.

2.2.1.1. Tipos de aprendizaje

Los algoritmos de aprendizaje automático se pueden clasificar en distintos tipos de algoritmos según el conocimiento a priori que se tiene en cada uno.

- **Aprendizaje supervisado.** En estos algoritmos se tiene un conjunto de datos previamente etiquetado y clasificado, es decir, un conjunto de muestra, el cual ya se sabe a qué grupo, valor o categoría pertenecen los ejemplos. Con este grupo de datos que llamamos datos de entrenamiento, se realiza el ajuste al modelo inicial planteado. Es de esta forma como el algoritmo va “aprendiendo” a clasificar las muestras de entrada

comparando el resultado del modelo y la etiqueta real de la muestra, realizando las compensaciones respectivas al modelo de acuerdo a cada error en la estimación del resultado.

- **Aprendizaje no supervisado.** En los algoritmos de aprendizaje no supervisado se trabaja de forma muy similar a los supervisados, con la diferencia de que éstos sólo ajustan su modelo predictivo tomando en cuenta los datos de entrada, sin importar los de salida. Es decir, a diferencia del supervisado, los datos de entrada no están clasificados ni etiquetados, y no son necesarias estas características para entrenar el modelo. Dentro de este tipo de algoritmos, el agrupamiento o clustering en inglés, es el más utilizado, ya que particiona los datos en grupos que posean características similares entre sí.
- **Aprendizaje semisupervisado.** Los algoritmos de aprendizaje semisupervisado se encuentran a medio camino entre el aprendizaje supervisado y el no supervisado, es decir poseen tanto datos etiquetados como datos no etiquetados. Un ejemplo de algoritmo de aprendizaje semisupervisado es un algoritmo de aprendizaje no supervisado al que se le añaden un conjunto de restricciones, como se va a hacer en este TFG para el problema del clustering.
- **Aprendizaje por refuerzo.** Los algoritmos de aprendizaje por refuerzo definen modelos y funciones enfocadas en maximizar una medida de “recompensas”, basados en “acciones” y al ambiente en el que el agente inteligente se desempeñará. Se busca que el algoritmo se ajuste a la mejor “recompensa” dada por el ambiente, y que su toma de acciones esté sujeta a estas recompensas.

2.2.1.2. **Redes Neuronales Artificiales**

Las redes neuronales artificiales son un modelo computacional de aprendizaje automático que consiste en un conjunto de unidades, llamadas neuronas artificiales, conectadas entre sí para transmitirse señales. La información de entrada atraviesa la red neuronal (donde se somete a diversas operaciones) produciendo unos valores de salida.

Cada neurona está conectada con otras a través de unos enlaces. En estos enlaces el valor de salida de la neurona anterior es multiplicado por un valor de peso. Estos pesos en los enlaces pueden incrementar o inhibir el estado de activación de las neuronas adyacentes. Del mismo modo, a la salida de la neurona, puede existir una función limitadora o umbral, que modifica el valor resultado o impone un límite que no se debe sobrepasar antes de propagarse a otra neurona. Esta función se conoce como función de activación.

Estos sistemas aprenden y se forman a sí mismos de forma automática, y por ello destacan en áreas donde la detección de soluciones o características es difícil de expresar con la programación convencional. Para realizar este aprendizaje automático, normalmente, se intenta minimizar una función de pérdida que evalúa la red en su total. Los valores de los pesos de las neuronas se van actualizando buscando reducir el valor de la función de pérdida. Este proceso se suele realizar mediante la propagación hacia atrás.

El objetivo de la red neuronal es resolver los problemas de la misma manera que el cerebro humano, aunque las redes neuronales son más abstractas, de hecho suelen contener desde

2. Definiciones y conceptos de computación

unos miles a unos pocos millones de unidades neuronales.

Cuando una red neuronal se usa para un problema de aprendizaje no supervisado vamos a distinguir el uso del aprendizaje competitivo que es el que usan los mapas auto-organizativos. El aprendizaje competitivo se basa en hacer que las neuronas pugnen entre sí para representar a una clase o patrón de entrada. La neurona seleccionada es aquella cuyos pesos incidentes se asemejan más al patrón de entrada. El aprendizaje consiste en reforzar las conexiones de la unidad ganadora y debilitar las otras, para que los pesos de la unidad ganadora se asemejen cada vez más al patrón de entrada. Luego, la reconstrucción de un patrón de entrada a partir de una neurona ganadora consiste en tomar los valores de peso de dicha neurona ya que son los que más se asemejan al patrón.

Parte II.

Mapas auto-organizativos (SOM) y Problema del agrupamiento con restricciones

3. El mapa auto-organizativo (SOM).

3.1. Cuantificación vectorial

3.1.1. Definición

La cuantificación vectorial se trata de un método clásico de aproximación que permite modelar funciones de densidad de probabilidad mediante la distribución de un conjunto de vectores de entrada.

Definición 3.1. Sea $X \subset \mathbb{R}^n$ un conjunto de vectores de \mathbb{R}^n e $Y = \{y_i \in \mathbb{R}^n : i = 1, 2, \dots, k\}$ un conjunto finito de vectores de \mathbb{R}^n . Se define la **función de cuantificación** como:

$$\begin{aligned} Q: X &\longrightarrow Y \\ x &\longmapsto y_c = Q(x), \end{aligned}$$

siendo

$$c = \arg \min_i \|x - y_i\|.$$

Por tanto, la cuantificación de x significa encontrar el vector $y_c \in Y$ más cercano a x con la distancia Euclídea de los que hay en Y . Más formalmente, se busca que:

$$\|x - y_c\| = \min_i \|x - y_i\|,$$

siendo $\|x - y_c\|$ el **error de cuantificación**.

Definición 3.2. Sea $y_i \in Y$, llamamos Región o **Conjunto de Voronoi** asociada a y_i a:

$$V_i = \{x \in \mathbb{R}^n : \|x - y_i\| \leq \|x - y_j\| \quad \forall j \neq i\}.$$

Definición 3.3. Como todo punto de \mathbb{R}^n queda asociado a algún y_i , se tiene que:

$$\bigcup_{i=1}^k V_i = \mathbb{R}^n,$$

y por tanto, $\{V_i : i \in 1, \dots, k\}$ es un recubrimiento finito de \mathbb{R}^n .

Como además

$$\bigcap_{i=1}^k \text{int}(V_i) = \emptyset,$$

se tiene que $\{V_i : i \in 1, \dots, k\}$ es una teselación que se denomina **Teselación de Voronoi**.

Definición 3.4. Una selección óptima del conjunto Y minimiza la denominada **distorsión de cuantificación**, que es una medida que se define como

$$E = \int \|x - y_c\|^2 p(x) dx,$$

3. El mapa auto-organizativo (SOM).

donde la integral esta tomada sobre todo el espacio métrico X y $p(x)$ es la función de densidad de x .

Se ha demostrado en [Ger79] que cuando los valores de Y minimizan E , también se minimiza el error medio de cuantificación en cada conjunto de Voronoi.

Nótese que el subíndice c es función de x y de todos los y_i con $i = 1, 2, \dots, k$; por lo que el gradiente de E con respecto a cualquiera de los y_i , no se puede formar fácilmente. De hecho, $\forall i \neq c$ si variamos y_i , el índice c puede dar un salto discontinuo del valor discreto c anterior a i , si $\|x - y_i\| < \|x - y_c\|$, pasándolo a ser $c = i$.

3.1.2. Derivación del algoritmo de Cuantificación Vectorial.

Voy a derivar el algoritmo, para obtener el conjunto Y que minimice E . Como no se dispone de ningún método exacto para la determinación del conjunto Y , al menos con un $p(x)$ general, voy a recurrir a un método de aproximación iterativo.

Suponiendo que $p(x)$ es continua. El problema causado por la discontinuidad de $c = c(x; y_1, \dots, y_k)$ se puede evitar si se hace uso de la siguiente identidad:

Proposición 3.1. Sea $\{a_i\}$ un conjunto de números escalares reales positivos, entonces en general se cumple que

$$\min_i a_i = \lim_{r \rightarrow -\infty} \left(\sum_i a_i^r \right)^{\frac{1}{r}}.$$

Otro resultado que se necesita es el siguiente:

Proposición 3.2. Tomando $f(x, r) = (1 + |x|^r)^{\frac{1}{r}}$. Se tiene que excluyendo todos los valores de x en los que f o $\lim_{r \rightarrow -\infty} f$ no son diferenciables, es decir, $x \in \{-1, 0, 1\}$, se cumple:

$$\lim_{r \rightarrow -\infty} \frac{\partial f}{\partial x} = \frac{\partial}{\partial x} \lim_{r \rightarrow -\infty} f.$$

Demostración. Reescribiendo la derivada como un límite:

$$\lim_{r \rightarrow -\infty} \frac{\partial f}{\partial x} = \lim_{r \rightarrow -\infty, h \rightarrow 0} \frac{f(x+h, r) - f(x, r)}{h}$$

y sustituyendo por el límite cuando $r \rightarrow -\infty$, se obtiene

$$\lim_{r \rightarrow -\infty, h \rightarrow 0} \frac{f(x+h, r) - f(x, r)}{h} = \lim_{h \rightarrow 0} \frac{\lim_{r \rightarrow -\infty} f(x+h) - \lim_{r \rightarrow -\infty} f(x)}{h}.$$

Entonces, en los puntos donde f y $\lim_{r \rightarrow -\infty} f$ son diferenciables, se tiene que

$$\lim_{h \rightarrow 0} \frac{\lim_{r \rightarrow -\infty} f(x+h) - \lim_{r \rightarrow -\infty} f(x)}{h} = \frac{\partial}{\partial x} \lim_{r \rightarrow -\infty} f.$$

□

3.1.2.1. Obtención del gradiente.

Para formar el gradiente $\nabla_{y_j} E$, se utiliza el hecho de que una función de la forma

$$g(x) = \left(\sum_i \|x - y_i\|^r \right)^{\frac{2}{r}}$$

es continua, univaluada, bien definida y continuamente diferenciable en sus argumentos, excepto cuando x es exactamente igual a algún y_i . Además tenemos que excluir los casos en que uno de los $\|x - y_i\|^r$ es exactamente igual a la suma de los otros términos. Pero como x es una variable aleatoria con función de densidad $p(x)$ continua, todos estos casos singulares tienen probabilidad cero.

Entonces se tiene por la **Proposición 3.1** que

$$\|x - y_c\|^2 = (\min_i \|x - y_i\|)^2 = \lim_{r \rightarrow -\infty} \left(\sum_i \|x - y_i\|^r \right)^{\frac{2}{r}}$$

y por la **Proposición 3.2**

$$\nabla_{y_j} E = \nabla_{y_j} \int \|x - y_c\|^2 p(x) dx = \int \lim_{r \rightarrow -\infty} \nabla_{y_j} \left(\sum_i \|x - y_i\|^r \right)^{\frac{2}{r}} p(x) dx.$$

Denotando ahora

$$A = \sum_i \|x - y_i\|^r,$$

se tiene que:

$$\nabla_{y_j} A^{\frac{2}{r}} = \frac{2}{r} A^{\frac{2}{r}-1} \nabla_{y_j} (\|x - y_j\|^r) = \frac{2}{r} A^{\frac{2}{r}-1} \nabla_{y_j} (\|x - y_j\|^2)^{\frac{r}{2}}.$$

Aplicando el gradiente y reordenando

$$\nabla_{y_j} A^{\frac{2}{r}} = -2A^{\frac{2}{r}} \frac{(\|x - y_j\|^2)^{\frac{r}{2}-1}}{A} (x - y_j).$$

Finalmente para calcular el límite de $\nabla_{y_j} A^{\frac{2}{r}}$, uso que

$$\lim_{r \rightarrow -\infty} A^{\frac{2}{r}} = \|x - y_c\|^2$$

y tomando

$$B = \frac{(\|x - y_j\|^2)^{\frac{r}{2}-1}}{A} = \frac{\|x - y_j\|^r}{\sum_i \|x - y_i\|^r} \|x - y_j\|^{-2} = \left(\sum_i \frac{\|x - y_i\|^r}{\|x - y_j\|^r} \right)^{-1} \|x - y_j\|^{-2},$$

3. El mapa auto-organizativo (SOM).

se tiene que cuando $r \rightarrow -\infty$, el término $\frac{\|x-y_i\|^r}{\|x-y_j\|^r}$, para cualquier y_j dado, es máximo cuando $y_i = y_c$, y empieza a predominar progresivamente sobre los otros términos; por lo tanto

$$\lim_{r \rightarrow -\infty} B = \lim_{r \rightarrow -\infty} \left(\frac{\|x-y_j\|}{\|x-y_c\|} \right)^r \|x-y_j\|^{-2} = \delta_{cj} \|x-y_j\|^{-2},$$

donde δ_{cj} es la delta de Kronecker.

Entonces combinando los resultados parciales obtenemos

$$\lim_{r \rightarrow -\infty} \nabla_{y_j} A^{\frac{2}{r}} = \lim_{r \rightarrow -\infty} -2A^{\frac{2}{r}} B(x-y_j) = -2\|x-y_c\|^2 \delta_{cj} \|x-y_j\|^{-2} (x-y_j) = -2\delta_{cj} (x-y_j),$$

y finalmente

$$\nabla_{y_j} E = \int \lim_{r \rightarrow -\infty} \nabla_{y_j} A^{\frac{2}{r}} p(x) d(x) = -2 \int \delta_{cj} (x-y_j) p(x) d(x).$$

La función muestral de este gradiente en el tiempo t es

$$\nabla_{y_j} E|_t = -2\delta_{cj} [x(t) - y_j(t)],$$

y el descenso más pronunciado en la trayectoria de optimización de E se produce en la dirección de $-\nabla_{y_j} E|_t$. Entonces, tras el cambio de índices, y la definición del “step size” por el factor $\alpha(t)$ que incluye la constante -2 , se obtiene

$$y_i(t+1) = y_i(t) + \alpha(t) \delta_{ci} [x(t) - y_i(t)].$$

3.2. El mapa auto-organizativo (SOM).

3.2.1. Definición de mapa auto-organizativo.

Un mapa auto-organizativo o auto-organizado (SOM) es un modelo de red neuronal que se utiliza para el aprendizaje no supervisado, por lo que no requiere el conocimiento de las etiquetas o valores objetivo. Fue introducido por Teuvo Kohonen en [Koh82b] y desarrollado con profundidad en [Koh90]. En la actualidad, se han publicado más de 4.000 artículos de investigación sobre él.

La teoría de la red SOM está motivada por la observación del funcionamiento del cerebro. Diversas impresiones sensoriales humanas se mapean neurológicamente en el cerebro, de forma que las relaciones espaciales o de otro tipo entre los estímulos se corresponden con las relaciones espaciales entre las neuronas.

Se trata de una herramienta de software eficaz para la visualización de datos de alta dimensión. En su forma básica, produce un grafo de similitud de los datos de entrada, así convierte las relaciones estadísticas no lineales entre los datos de alta dimensión en simples relaciones geométricas entre la imagen de sus puntos en el espacio de baja dimensión. A este espacio de baja dimensión se le va a denominar mapa, y a sus elementos neuronas.

El objetivo principal de un mapa auto-organizativo es extraer información oculta o patrones comunes de los datos sin la necesidad de la utilización de datos etiquetados que sirvan

como ejemplo. Esto lo consigue al mapear los datos de entrada de un espacio n -dimensional al espacio de menor dimensión o mapa que ya se ha introducido. En este conjunto de menor dimensión se siguen manteniendo las relaciones topológicas originales, y por tanto la ubicación de los puntos en el mapa muestra la similitud relativa entre los puntos en el espacio multidimensional. Como el SOM comprime la información al tiempo que preserva las relaciones topológicas y/o métricas más importantes de los elementos, también produce un tipo de abstracción que es válida para utilizarse en varias tareas complejas como el aprendizaje automático.

3.2.2. Formalización.

Sea $X = \{x_1, \dots, x_m\}$ el conjunto de posibles variables de entrada, donde $x_i \in \mathbb{R}^n$ con $i = 1, \dots, m$. Y sea $Y = \{y_1, \dots, y_k\}$ el conjunto de neuronas que forma el mapa. Se tiene que la dimensión de las neuronas tiene que ser la misma que la de los datos de entrada, por tanto, $y_j \in \mathbb{R}^n$ con $j = 1, \dots, k$. Además este conjunto describe un grafo, tomando como nodos los elementos y_i y como arcos una relación de proximidad entre las neuronas inducida por una función de vecindad que define la topología de Y . Este grafo con sus nodos y relaciones es lo que se denomina como mapa auto-organizativo.

El SOM entonces lleva a cabo un mapeo no lineal, ordenado y suave que se implementa de la siguiente manera y que se asemeja a la cuantificación vectorial clásica (Sección 3.1).

Sea $X = \{x_1, \dots, x_m\}$ un conjunto, donde cada elemento $x_i \in X \subset \mathbb{R}^n$. Y sea $Y = \{y_1, \dots, y_k\}$ otro conjunto con $y_j \in Y \subset \mathbb{R}^n$. Podemos definir el mapeo, como:

$$\begin{aligned} BMU : X &\rightarrow Y \\ x &\rightarrow y_c = BMU(x), \end{aligned}$$

con

$$c = \arg \min_i d(x, y_i),$$

siendo $d(x, y_i)$ la distancia Euclídea e y_c la neurona más cercana a x con esa métrica. (La notación de BMU viene del inglés de “Best Matching Unit”).

A diferencia de la cuantificación vectorial clásica, nuestra tarea consiste en definir el conjunto de neuronas $\{y_i : i = 1, \dots, k\}$ de tal manera que el mapeo sea ordenado y descriptivo de la distribución de x . Una opción puede ser inicializar las neuronas de forma aleatoria, preferiblemente del dominio de las muestras de entrada.

Tomando x un vector aleatorio, se podría decir entonces que el SOM es una “proyección no lineal” de la función de densidad de probabilidad $p(x)$ del vector de datos de entrada sobre el mapa. Podríamos considerar un “mapeo óptimo” aquel que proyecta la función de densidad de probabilidad $p(x)$ de la manera más “fiel”, tratando de preservar al menos las estructuras locales de $p(x)$ en el mapa de salida.

3. El mapa auto-organizativo (SOM).

3.2.3. Entrenamiento del SOM

En esta sección voy a explicar el procedimiento que se lleva a cabo en el entrenamiento de un mapa auto-organizativo.

Se toma una lista de muestras de entrada x y en primer lugar se inicializan los valores de las neuronas del mapa de forma que tengan la misma dimensión que los datos de entrada. Considerando t como el índice de la lista, se compara cada $x(t)$ con todos los y_i y se copia cada x en la sublista asociada a la neurona más similar a $x(t)$, como se aprecia en la **Figura 3.1**. Cuando todas las $x(t)$ se han distribuido en las respectivas sublistas de esta manera, se considera el conjunto de vecindad N_i alrededor de la neurona y_i , donde N_i consiste en el conjunto de todos los nodos que se sitúan dentro de una circunferencia de un cierto radio y con centro en y_i . Entonces, las neuronas de la red ajustan sus pesos basándose en las sublistas de entradas que existen en su conjunto de vecindad.

El procedimiento anterior se itera dando lugar al aprendizaje de la red, es decir, se vuelven a pasar como entrada todas las $x(t)$ originales, que se volverán a asignar en las sublistas. Estas sublistas cambiarán de una iteración a otra puesto que el valor de las neuronas cambia con el ajuste.

Durante el aprendizaje, aquellos nodos que están topológicamente cerca en el grafo hasta una cierta distancia, se activarán mutuamente para aprender algo de la misma entrada x . Esto dará lugar a una relajación local o efecto de suavizado en el vector de pesos de las neuronas en esta vecindad, que en el aprendizaje continuado conduce a un ordenamiento global.

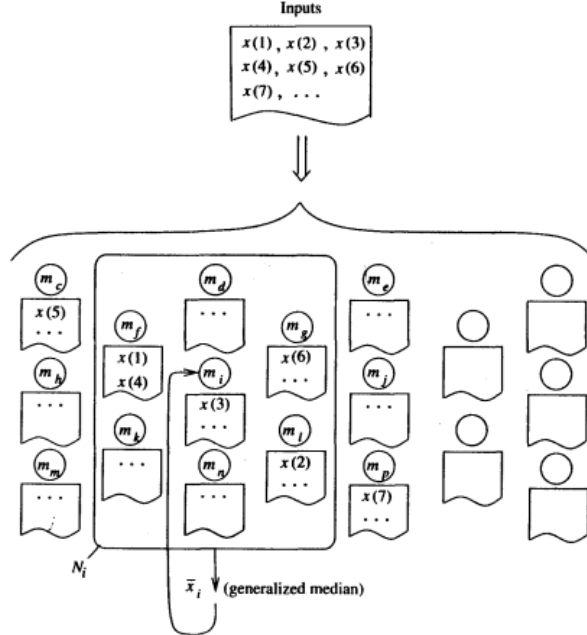


Figura 3.1.: Mapa bidimensional de neuronas m_i con sus sublistas de instancias asociadas.

3.2.3.1. Convergencia de las neuronas.

Consideremos los eventuales límites de convergencia del siguiente proceso de aprendizaje, en el que los valores iniciales de las neuronas, $\{y_i(0) : i = 1, \dots, k\}$, pueden ser arbitrarios, por ejemplo, aleatorios:

$$y_i(t+1) = y_i(t) + h_{ci}(t)[x(t) - y_i(t)],$$

donde $t = 0, 1, 2, \dots$ es un número entero al que nos referimos como la coordenada de tiempo discreto. Y h es la función de vecindad, que se define como:

$$h_{ci}(t) = h(\|r_c - r_i\|, t),$$

donde $r_c, r_i \in \mathbb{R}^n$ son la posición en el mapa de las neuronas y_c e y_i respectivamente. Además se tiene que cuando $\|r_c - r_i\|$ aumenta, $h_{ci} \rightarrow 0$.

El proceso de aprendizaje del SOM producirá probablemente valores asintóticamente convergentes para las neuronas y_i , cuya colección se aproximará a la distribución de las muestras de entrada $x(t)$, incluso de forma ordenada.

3.2.3.2. Función de vecindad.

En el proceso de relajación, la función h_{ci} tiene un papel muy importante pues define un núcleo de suavización sobre los puntos de la red. Además para la convergencia del proceso de aprendizaje es necesario que $h_{ci}(t) \rightarrow 0$ cuando $t \rightarrow \infty$.

En la literatura, hay dos opciones simples para la función h que aparecen con frecuencia. La más sencilla de ellas se refiere a un conjunto de vecindad de puntos alrededor del nodo c . Dejemos que su conjunto de índices se denote N_c (nótese que podemos definir $N_c = N_c(t)$ como una función del tiempo), con lo que

$$h_{ci}(t) = \begin{cases} \alpha(t) & \text{si } i \in N_c \\ 0 & \text{si } i \notin N_c \end{cases}.$$

Al valor $\alpha(t)$ se le denomina tasa de aprendizaje con $0 < \alpha(t) < 1$. Tanto $\alpha(t)$ como el radio de N_c suelen ser monótonamente decrecientes respecto al tiempo (durante el proceso de ordenación).

Otro núcleo de vecindad más suave y ampliamente aplicado puede escribirse en términos de la función Gaussiana,

$$h_{ci}(t) = \alpha(t) \cdot \exp\left(-\frac{\|r_c - r_i\|^2}{2\sigma^2(t)}\right),$$

donde $\alpha(t)$ vuelve a ser la tasa de aprendizaje, y el parámetro $\sigma(t)$ define la anchura del núcleo; este último corresponde al radio del N_c anterior. Tanto $\alpha(t)$ como $\sigma(t)$ son funciones monótonamente decrecientes respecto al tiempo. Dos ejemplos de vecindarios topológicos se muestran en la [Figura 3.2](#).

3. El mapa auto-organizativo (SOM).

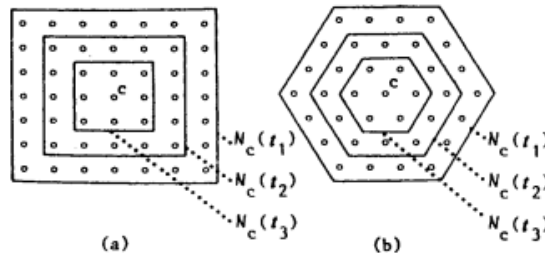


Figura 3.2.: Dos ejemplos de vecindarios topológicos.

Finalmente, se va a describir en **Algoritmo 1** el algoritmo SOM básico con pseudo-código.

	Algorithm 1: SOM básico
	Input : conjunto de datos X , dimensión del mapa k , número de épocas n
1	# Se inicializan aleatoriamente las neuronas del mapa $Y = \{y_1, y_2, \dots, y_k\}$
2	# con la misma dimensión que la entrada y se establece un factor de.
3	# aprendizaje inicial α .
4	som := RandInit(X, k, α);
5	for $i := 0$ hasta n do
6	# Se elige al azar un vector de entrada $x(t)$.
7	$x := \text{Rand}(X)$;
8	# Se obtiene la neurona más cercana a $x(t)$.
9	$y := \text{som.BMU}(x)$;
10	# Se ajustan los pesos de las neuronas en la vecindad de la BMU
11	# acercando los valores al valor del vector de entrada.
12	som.Adjust(y, x);
13	# Se actualiza α y se disminuye el radio del vecindario.
14	som.UpdateParams();
15	end
16	return som.solution

3.2.4. Visualización del proceso de adaptación.

Con tal de visualizar como es el proceso de aprendizaje y adaptación de un mapa auto-organizativo, he creado dos conjuntos de datos sencillos que se pueden mostrar en dos y tres dimensiones. Entonces para cada uno de ellos entrenaré un mapa auto-organizado durante un gran número de épocas n . Además con tal de mostrar ejemplos de diferentes topologías de mapa, entrenaré el primer conjunto con un SOM bidimensional de tamaño 4×4 y el segundo con un SOM tridimensional de tamaño $2 \times 2 \times 2$. La adaptación del SOM 2D se muestra en la **Figura 3.3** y la del SOM 3D en la **Figura 3.4**.

3.2. El mapa auto-organizativo (SOM).

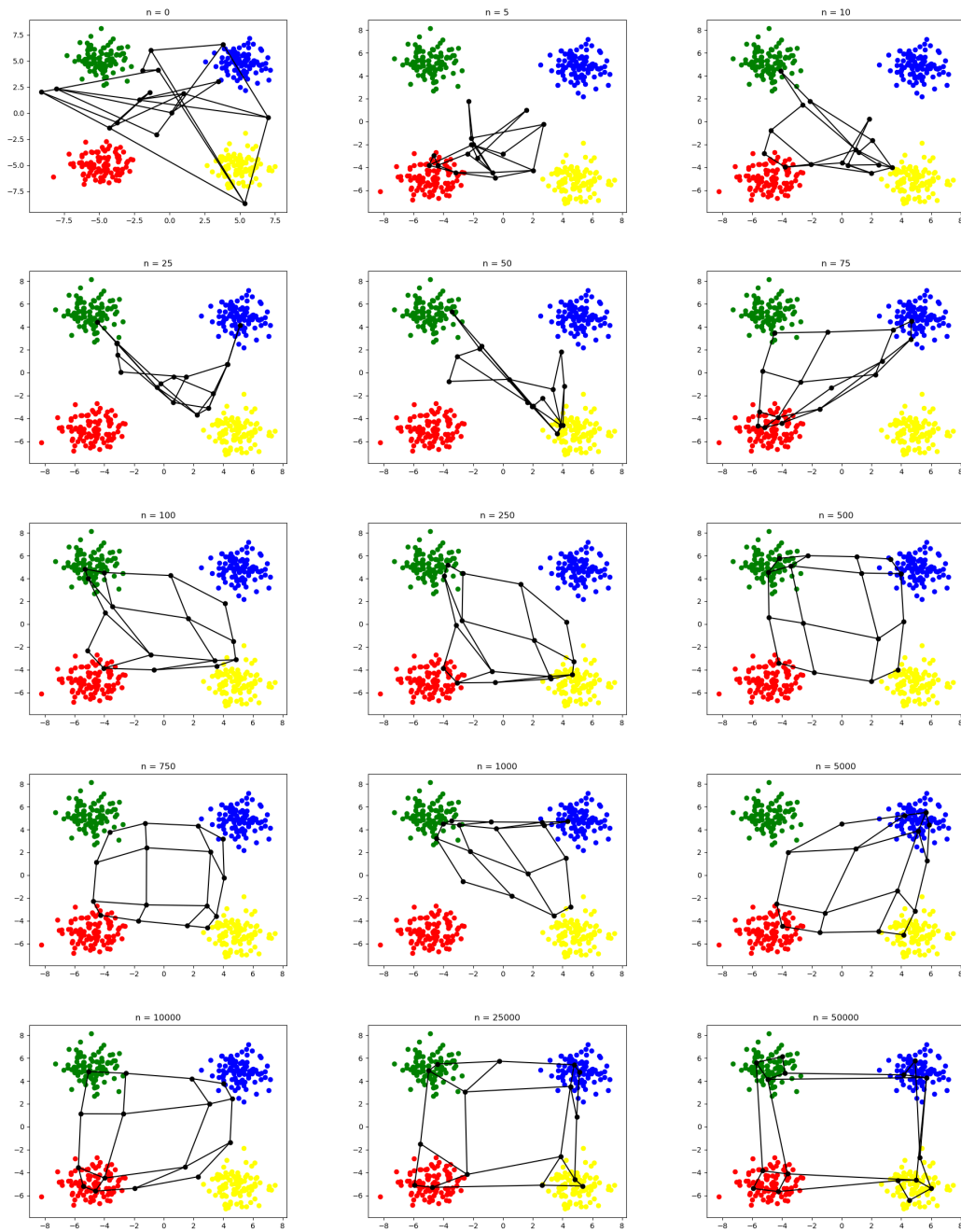


Figura 3.3.: Adaptación del SOM 2D.

3. El mapa auto-organizativo (SOM).

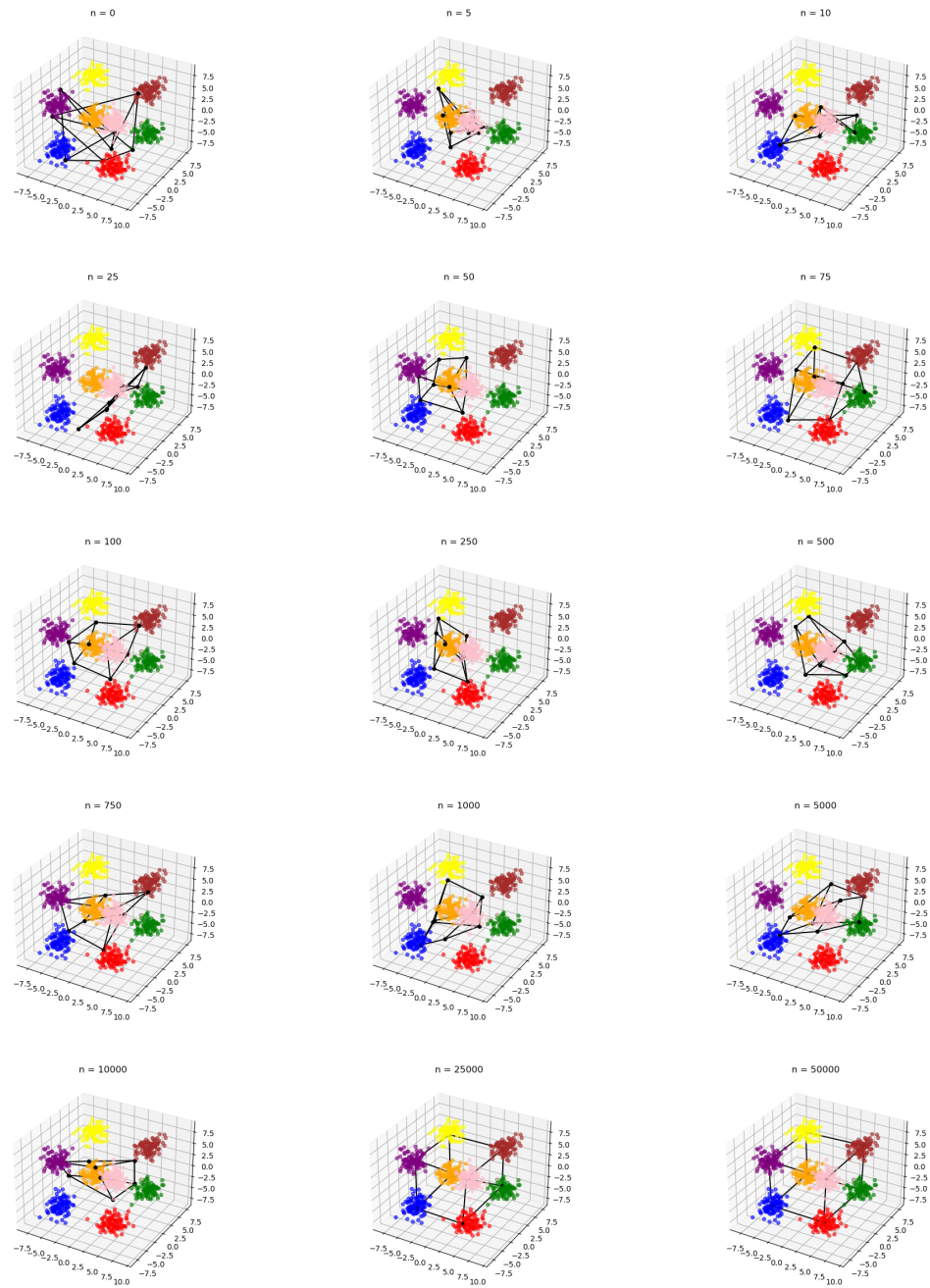


Figura 3.4.: Adaptación del SOM 3D.

3.2.5. Variante del SOM

Voy a presentar una variante del SOM que será la que usaré en la implementación de los algoritmos para el clustering con restricciones. Esta variante se caracteriza porque en cada época se entrena al SOM con todas las instancias del conjunto de entradas X en vez de con una sola elegida al azar. Para darle aleatoriedad al orden de las instancias, se baraja el conjunto de entradas antes de comenzar cada época. La variante se explica con más detalle con pseudo-código en [Algoritmo 2](#).

Esta variante del SOM será muy útil para el problema del clustering con restricciones, aunque su justificación la dejaremos para la [Sección 6.1](#).

Algorithm 2: Variante del SOM

```

Input : conjunto de datos  $X$ , dimensión del mapa  $k$ , número de épocas  $n$ 
1 # Se inicializan aleatoriamente las neuronas del mapa  $Y = \{y_1, y_2, \dots, y_k\}$ 
2 # con la misma dimensión que la entrada  $y$  se establece un factor de.
3 # aprendizaje inicial  $\alpha$ .
4  $\text{som} := \text{RandInit}(X, k, \alpha)$ ;
5 for  $i := 0$  hasta  $n$  do
6   # Se baraja de forma aleatoria el conjunto de entradas.
7    $X := \text{RandomShuffle}(X)$ ;
8   # Se entrena el mapa con todas las entradas del conjunto.
9   for  $j := 0$  hasta  $\text{tam}(X)$  do
10    # Se obtiene la neurona más cercana a  $x_j$ .
11     $y := \text{som.BMU}(x[j])$ ;
12    # Se ajustan los pesos de las neuronas en la vecindad de la BMU
13    # acercando los valores al valor del vector de entrada.
14     $\text{som.Adjust}(y, x[j])$ ;
15    # Se actualiza  $\alpha$  y se disminuye el radio del vecindario.
16     $\text{som.UpdateParams}()$ ;
17   end
18 end
19 return  $\text{som.solution}$ 

```

3.3. Fundamentos Matemáticos del SOM

La esencia de las matemáticas no es hacer las cosas simples complicadas, sino hacer las cosas complicadas simples.

(Stanley Gudder)

3.3.1. Proceso de Markov

Definición 3.5. Se define un **proceso estocástico** como una colección de variables aleatorias definidas en un espacio de probabilidad común

$$\{X(t) : t \in T\},$$

con

$$X(t) : (\Omega, \mathcal{A}, P) \rightarrow (\mathbb{R}, \mathcal{B}).$$

En la mayoría de problemas $t \in T$ tiene el significado de tiempo, por lo que $X(t)$ es una variable aleatoria que representa un valor observado en el tiempo t .

Definición 3.6. Se denomina como **propiedad de Markov** a la propiedad de ciertos procesos estocásticos por la cual “carecen de memoria”, lo que significa que la distribución de probabilidad del valor futuro de una variable aleatoria depende únicamente de su valor presente, siendo independiente de la historia de dicha variable.

Definición 3.7. Un **proceso de Markov** es un fenómeno aleatorio que cumple la propiedad de Markov.

Consta de los siguientes elementos:

- **Un espacio de estados** $E = \{a_1, a_2, a_3, \dots\}$. Cada estado son todos los resultados posibles de cada una de las variables aleatorias). Cuando el resultado en el tiempo t es a_i decimos que el proceso está en el estado a_i en el tiempo t -ésimo.
- **Probabilidad de transición.** Para cada dos estados a_i y a_j , a la probabilidad de que el estado a_j ocurra inmediatamente después que el estado a_i se le denomina probabilidad de transición del estado a_i al estado a_j , y se designa por P_{ij} .
- **Matriz de transición.** Matriz con todas las probabilidades de transición P_{ij} .
- **Vector de probabilidad inicial.** Vector formado por las probabilidades del estado inicial del proceso.

Dentro del conjunto de estados, definimos diferentes tipos de estados:

- **Estado transitorio.** Estados que pueden sucederse a sí mismos y, además, es posible alcanzar, por lo menos, alguno de los restantes desde ellos.
- **Estado absorbente.** Estado tal que si el proceso entra en él permanecerá indefinidamente en este estado (ya que las probabilidades de pasar a cualquiera de los otros son cero).

Un proceso de Markov que consta de estados transitorios y absorbentes se dice que es un proceso absorbente de Markov.

Definición 3.8. Si un proceso de Markov contiene algún estado absorbente, se le denomina **proceso absorbente de Markov**, y la línea de la matriz de transición correspondiente a las probabilidades de transición de dicho estado constará de un 1 en la diagonal principal y ceros en los demás elementos. Si dicho estado, partiendo de un estado inicial arbitrario, se alcanza mediante alguna secuencia de entradas que tenga una probabilidad positiva, entonces, permitiendo una secuencia aleatoria de entradas, el estado absorbente se alcanza con probabilidad uno, cuando $t \rightarrow \infty$.

3.3.2. Auto-organización

Aunque el principio básico del SOM parece sencillo, el comportamiento del proceso, especialmente en relación con las representaciones de entrada más complejas, ha sido difícil de describir en términos matemáticos. La primera aproximación que se hace a continuación analiza el proceso en su forma más simple, pero parece que también se pueden obtener resultados fundamentalmente similares con sistemas más complejos.

3.3.2.1. Caso 1 dimension

Intentaré justificar la capacidad de autoorganización de forma analítica, utilizando un modelo de sistema muy sencillo. Las razones de los fenómenos de autoorganización son en realidad muy sutiles y sólo se han demostrado estrictamente en los casos más simples. En esta sección voy a dar una explicación básica del proceso de Markov que debería ayudar a entender la naturaleza del proceso.

En primer lugar, limitaremos nuestras consideraciones a un vector de neuronas de una dimensión.

Sea $\epsilon \in \mathbb{R}$ e $Y = \{y_i \in \mathbb{R} : i = 1, 2, \dots, l\}$. El Proceso de Adaptación de Y a ϵ tendrá los siguientes pasos:

1. **Encontrar el y_i más similar a ϵ , y_c .** La similitud entre ϵ y y_i se define por el valor absoluto de su diferencia $|\epsilon - y_i|$; la mejor coincidencia se define por

$$|\epsilon - y_c| = \min_i |\epsilon - y_i|.$$

2. **Definición del vecindario de y_c .** Defino el vecindario N_c , como el conjunto de y_i que se actualizan al elegir y_c como mejor coincidencia. Concretamente vamos a definir el vecindario como:

$$N_c = \{\max(1, c - 1), c, \min(l, c + 1)\}.$$

En otras palabras, y_i tiene como vecinos y_{i-1} e y_{i+1} , excepto en los extremos del vector Y , donde el único vecino de y_1 es y_2 , y el único vecino de y_l es y_{l-1} . Entonces N_c es simplemente el conjunto formado por y_c y sus vecinos inmediatos.

3. El mapa auto-organizativo (SOM).

3. **Definición del $\alpha > 0$ y adaptación.** La naturaleza general del proceso es similar para diferentes valores de $\alpha > 0$; que es principalmente la velocidad a la que el proceso varía. En el formalismo de tiempo continuo, las ecuaciones son

$$\begin{aligned} \frac{\partial y_i}{\partial t} &= \alpha(\epsilon - y_i) & \text{con } i \in N_c \\ \frac{\partial y_i}{\partial t} &= 0 & \text{con } i \notin N_c \end{aligned}.$$

Proposición 3.3. Sea $\epsilon \in \mathbb{R}$ una variable aleatoria e $Y = \{y_i(t) \in \mathbb{R} : i = 1, 2, \dots, l\}$, donde cada y_i ha sido inicializada con un valor aleatorio. El conjunto de las y_i asumirán gradualmente nuevos valores por el proceso especificado anteriormente, de manera que:

1. Cuando $t \rightarrow \infty$, el conjunto Y se ordena en una secuencia ascendente o descendente. Y una vez que el conjunto está ordenado, permanece así $\forall t$.
2. La función de densidad puntual de las y_i se aproximará finalmente a alguna función monótona de la función de densidad de probabilidad $p(\epsilon)$ de ϵ .

La demostración de la proposición se llevará a cabo en dos partes: formación de secuencias ordenadas de las y_i , y su convergencia a ciertos "puntos fijos", respectivamente.

3.3.2.2. Ordenación de Y

Proposición 3.4. En el proceso de adaptación definido anteriormente, las y_i se ordenan con probabilidad uno en orden ascendente o descendente cuando $t \rightarrow \infty$.

Demostración. Sea ϵ una variable aleatoria con soporte finito y función de densidad $p(\epsilon)$. Y considerando $\epsilon = \epsilon(t) \in \mathbb{R}$ como un valor aleatorio, con $\epsilon(t_1)$ y $\epsilon(t_2)$ independientes para cualquier $t_1 \neq t_2$.

La prueba se basa en las propiedades generales de los procesos de Markov, especialmente la del estado absorbente. En primer lugar se identifica el estado absorbente con cualquiera de las secuencias ordenadas de las y_i , (tanto la ascendente como la descendente, que juntas constituyen el estado absorbente). Y si con un estado inicial arbitrario este estado se alcanza mediante alguna secuencia de entradas que tenga una probabilidad positiva, entonces, permitiendo una secuencia aleatoria de entradas, el estado absorbente se alcanza también con probabilidad uno, cuando $t \rightarrow \infty$.

En primer lugar tenemos que tomando $\forall t, \epsilon(t) \in [a, b] \subset \mathbb{R}$, se tiene que

$$\exists n \text{ tal que } \forall t \geq n : y_i(t) \in [a, b] \quad \forall i \in \{1, 2, \dots, l\},$$

ya que las y_i se adaptan acercándose a ϵ , y por tanto a $[a, b]$ progresivamente.

Igualmente tomando $\epsilon(t) \in [a, b]$, y considerando y_c como la y_i más similar, entonces y_{c-1} , y_c e y_{c+1} se adaptarán a $\epsilon(t)$ por pertenecer al vecindario de y_c . De esta forma se acercan y_{c-1} e y_{c+1} a y_c . Entonces si hay desorden a algún lado de y_c , como por ejemplo, si $y_{c-1} < y_{c-2} < y_c$ e y_{c-2} se mantiene constante,

$$\exists n \text{ tal que } \forall t \geq n : y_{c-1} \in [y_{c-2}, y_c] \quad \forall c \in \{3, \dots, l\}.$$

Equivalentemente ante un desorden por la derecha se tiene que

$$\exists n \text{ tal que } \forall t \geq n : y_{c+1} \in [y_c, y_{c+2}] \forall c \in \{1, \dots, l-2\}.$$

Por tanto, se tiene que si tomamos un número de ϵ cercanos a una y_i , hará que y_{i-1} al acercarse a y_i se sitúe entre y_{i-2} e y_i (ya que y_{i-2} no se modifica).

De esta forma se tiene que el vector de ordenará globalmente en un número finito de pasos tal que:

$$\exists n \text{ tal que } \forall t \geq n : y_c \in [y_{c-1}, y_{c+1}] \forall c \in \{2, \dots, l-1\}.$$

□

Corolario 3.1. *Si todos los valores están ordenados, no pueden desordenarse en una actualización posterior.*

Demostración. La prueba se deduce directamente de la observación de que si todas las secuencias parciales están ordenadas, entonces el proceso de adaptación no puede cambiar el orden relativo de ningún par (y_i, y_j) , $i \neq j$. □

3.3.2.3. Fase de convergencia

Una vez que las y_i se han ordenado, su convergencia final a los valores asintóticos es de especial interés, ya que estos últimos representan la imagen de la distribución de entrada $p(\epsilon)$.

En esta subsección se asume de ahora en adelante que Y está ordenado y, a causa del **Corolario 3.1** que, sigue estándolo en posteriores procesos de actualización.

El objetivo es calcular los valores asintóticos de las y_i . Estos valores asintóticos se obtienen en el sentido de mínimos cuadrados o de la convergencia casi segura sólo si el “factor de aprendizaje” $\alpha = \alpha(t)$ in (3.9) tiende a cero; la secuencia $\{\alpha(t) | t = 0, 1, \dots\}$ debe obviamente satisfacer ciertas condiciones similares a las impuestas en un proceso de aproximación estocástica.

Las propiedades de convergencia de las y_i se discuten en un sentido menos restringido, es decir, sólo se analiza el comportamiento dinámico de los valores esperados $E\{y_i\}$. Se demostrará que estos números convergen a límites únicos. Las varianzas de las y_i pueden hacerse arbitrariamente pequeñas mediante una elección adecuada de $\alpha(t)$ cuando $t \rightarrow \infty$.

Partiendo del ordenamiento de las y_i , y como un nodo seleccionado sólo puede afectar a sus vecinos inmediatos, es obvio a partir del Proceso de Adaptación que cualquier valor particular de y_i puede ser afectado sólo si ϵ llega a un intervalo S_i definido de la siguiente

3. El mapa auto-organizativo (SOM).

manera (suponiendo $l \geq 5$):

$$\begin{aligned} S_1 &= \left[a, \frac{1}{2} (y_2 + y_3) \right]. \\ S_2 &= \left[a, \frac{1}{2} (y_2 + y_3) \right]. \\ S_i &= \left[\frac{1}{2} (y_{i-2} + y_{i-1}), \frac{1}{2} (y_{i+1} + y_{i+2}) \right] \quad \text{si } 3 \leq i \leq l-2. \\ S_{l-1} &= \left[\frac{1}{2} (y_{l-3} + y_{l-2}), b \right]. \\ S_l &= \left[\frac{1}{2} (y_{l-2} + y_{l-1}), b \right]. \end{aligned} \quad (3.1)$$

Los valores esperados de las $\frac{\partial y_i}{\partial t} = \dot{y}_i$, condicionados a las y_i , según (3.9) son

$$\langle \dot{y}_i \rangle = E[\dot{y}_i] = \alpha (E[\epsilon | \epsilon \in S_i] - y_i) P[\epsilon \in S_i],$$

donde $P[\epsilon \in S_i]$ es la probabilidad de que ϵ caiga en el intervalo S_i . Ahora $E[\epsilon | \epsilon \in S_i]$ es el centro de gravedad de S_i (ver (3.1)), que es una función de la y_i cuando $p(\epsilon)$ ha sido definida. Para resolver el problema de una forma cercana simplificada, se supone que $p(\epsilon) \equiv \text{constante}$ sobre el soporte $[a, b]$ y cero fuera de él, con lo que primero se obtiene que:

$$\begin{aligned} \langle \dot{y}_1 \rangle &= \frac{\alpha}{4} (2a + y_2 + y_3 - 4y_1) P[\epsilon \in S_1]. \\ \langle \dot{y}_2 \rangle &= \frac{\alpha}{4} (2a + y_3 + y_4 - 4y_2) P[\epsilon \in S_2]. \\ \langle \dot{y}_i \rangle &= \frac{\alpha}{4} (y_{i-2} + y_{i-1} + y_{i+1} + y_{i+2} - 4y_i) P[\epsilon \in S_i] \quad \text{si } 3 \leq i \leq l-2. \\ \langle \dot{y}_{l-1} \rangle &= \frac{\alpha}{4} (y_{l-3} + y_{l-2} + 2b - 4y_{l-1}) P[\epsilon \in S_{l-1}]. \\ \langle \dot{y}_l \rangle &= \frac{\alpha}{4} (y_{l-2} + y_{l-1} + 2b - 4y_l) P[\epsilon \in S_l]. \end{aligned} \quad (3.2)$$

Partiendo de condiciones iniciales arbitrarias $y_i(0)$, las trayectorias más probables “promediadas”, $y_i(t)$, se obtienen como soluciones de una ecuación diferencial equivalente correspondiente a la ecuación (3.2)

$$\frac{\partial z}{\partial t} = P(z)(Fz + h) \quad (3.3)$$

con

$$z = [y_1, y_2, \dots, y_l]^T,$$

$$F = \frac{\alpha}{4} \begin{bmatrix} -4 & 1 & 1 & 0 & 0 & 0 & 0 & \dots \\ 0 & -4 & 1 & 1 & 0 & 0 & 0 & \\ 1 & 1 & -4 & 1 & 1 & 0 & 0 & \\ 0 & 1 & 1 & -4 & 1 & 1 & 0 & \\ \vdots & & & & \ddots & & & \vdots \\ & & & & & 0 & 1 & 1 & -4 & 1 & 1 & 0 \\ & & & & & 0 & 0 & 1 & 1 & -4 & 1 & 1 \\ & & & & & 0 & 0 & 0 & 1 & 1 & -4 & 0 \\ \dots & & & & & 0 & 0 & 0 & 0 & 1 & 1 & -4 \end{bmatrix},$$

$$h = \frac{\alpha}{2} [a, a, 0, 0, \dots, 0, b, b]^T,$$

y

$$P(z) = \{d_{ii} = P(\epsilon \in S_i)\}$$

matriz diagonal, siendo d_{ii} sus elementos en la diagonal.

La ecuación (3.3) es una ecuación diferencial de primer orden con coeficientes constantes. Tiene un punto fijo de solución, una solución particular con $\frac{\partial z}{\partial t}$, que es (teniendo en cuenta que $P(z)$ es diagonal y definida positiva)

$$z_0 = -F^{-1}h,$$

siempre que exista F^{-1} ; esto ha sido demostrado estrictamente por Kohonen en [Koh82b].

La solución general de la ecuación (3.3) es difícil de obtener. Algunos estudios recientes, como [CF87], indican que la convergencia al punto fijo es generalmente cierta, por lo que restringimos nuestras consideraciones a mostrar sólo la solución asintótica.

4. Problema del agrupamiento con restricciones

4.1. Descripción del problema

El problema a tratar consiste en realizar agrupamientos (también llamados clusters) entre las instancias de un conjunto de datos de forma que se agrupen las que sean más similares entre sí, pero con la consideración de algunas restricciones establecidas para ciertos pares de instancias del conjunto. Concretamente estas restricciones se dividen en dos tipos, ML (Must-Link) si estos pares deben estar agrupados en el mismo cluster, y CL (Cannot-Link), si deben estar agrupados en distintos clusters.

Se tiene un conjunto de datos: $X = \{x_1, \dots, x_n\}$, formado por n instancias, y debemos de encontrar una partición $\mathcal{C} = \{C_1, \dots, C_k\}$ del mismo, de forma que:

- $\bigcup_{i=1}^k C_i = X$,
- $C_i \neq \emptyset \forall i \in \{1, \dots, k\}$, y
- $C_i \cap C_j = \emptyset \forall i, j \in \{1, \dots, k\}$ con $i \neq j$.

Y que además minimice la desviación general y cumpla con el conjunto de restricciones R . Al conjunto de clusters resultante se le denomina configuración de clusters.

Las restricciones se establecen o bien en una matriz de dimensión $n \times n$ o bien por dos conjuntos de parejas de instancias, uno para restricciones ML (must-link) y otro para restricciones CL (cannot-link). Los conjuntos se denominarán ML y CL y las parejas son de la forma (x_i, x_j) con $x_i, x_j \in X$. Por tanto, se tiene que el conjunto de todas las restricciones R , se obtiene como

$$R = ML \cup CL.$$

En el caso de la representación como matriz, esta será simétrica y contendrá para cada par de instancias x_i y x_j , el valor de su restricción, en la posición ij de la matriz, que será 1 (ML), -1 (CL) o 0 (si no existe restricción), ya que hay que tener en cuenta que no todas las parejas suelen tener restricciones.

Para trabajar en este problema usaré los siguientes términos:

- Para cada cluster puedo definir su **centroide** asociado como:

$$\mu_i = \frac{1}{|C_i|} \sum x_j \text{ con } x_j \in C_i.$$

- Para cada cluster puedo definir su **distancia media intra-cluster** como:

$$\bar{C}_i = \frac{1}{|C_i|} \sum \|x_j - \mu_i\| \text{ con } x_j \in C_i.$$

4. Problema del agrupamiento con restricciones

- La **desviación general** de la partición se define como:

$$\bar{C} = \frac{1}{k} \sum \bar{C}_i.$$

- La **función de clustering**, π , es la función que asigna a cada instancia el cluster al que pertenece. Se define como:

$$\begin{aligned} \pi : X &\rightarrow \mathcal{C} \\ x_i &\rightarrow C_j = \pi(x_i), \end{aligned}$$

siendo C_j el subconjunto de la partición \mathcal{C} al que x_i pertenece.

- La **infactibilidad** es el número de restricciones que incumple una partición.

Tomando $x_i = ML_{i,1}$, $C_r = \pi(ML_{i,2})$, $y_j = CL_{j,1}$ y $C_s = \pi(CL_{j,2})$, se tiene que

$$infactibilidad(\mathcal{C}) = \sum_{i=0}^{|ML|} \chi_{X-C_r}(x_i) + \sum_{j=0}^{|CL|} \chi_{C_s}(y_j),$$

donde $X - c_r$ es el complementario al conjunto C_r en el conjunto X , χ es la función característica y, con $|ML|$ y $|CL|$ nos referimos a la cardinalidad de esos conjuntos.

4.2. Estudio de la complejidad

Dado que el clustering restringido añade un nuevo elemento, las restricciones, al problema de clustering, debemos considerar cómo afectan a la complejidad del problema. El problema de viabilidad para el clustering con restricciones a nivel de instancia se definió en [DR05] como puede verse en la Def. 4.1.

Definición 4.1. Se define el **problema de viabilidad** para el clustering con restricciones como: dado un conjunto de datos X , un conjunto de restricciones CS y un número de clusters k , ¿existe una partición C de X con k clusters tal que se satisfagan todas las restricciones en CS ? [DR05]

En [DR05] se demuestra que el problema de viabilidad para el clustering con restricciones es *NP-completo*, reduciéndolo a partir del problema de K-Colorabilidad del grafo (también se demuestra que no es más difícil, por lo que ambos tienen la misma complejidad). La Tabla 4.1 muestra la complejidad de la viabilidad para diferentes tipos de restricciones. Estos resultados de complejidad muestran que el problema de viabilidad con restricciones CL es intratable y, por tanto, la agrupación restringida también es intratable. Los problemas intratables son difíciles de resolver con métodos deterministas y exactos. Por esta razón, abordar este problema desde el aprendizaje automático es un buen enfoque para encontrar una solución de calidad para el problema.

Tipo de restricciones	Complejidad
Must-Link	P
Cannot-Link	NP-completo
ML y CL	NP-completo

Tabla 4.1.: Complejidad del problema.

4.3. Métodos desarrollados para el problema

Para buscar una resolución del problema se necesitan una serie de cálculos que van a permitir medir la calidad de la solución. Estos son el cálculo de la infactibilidad, de los centroides y de la desviación general de la partición; y para esta última se van a hacer uso de distancias. Estos cálculos se implementarán en varias funciones que se explicarán a continuación en pseudo-códigos. Las funciones que calculan las distancias se definirán en [Algoritmo 3](#) y [Algoritmo 4](#); y las que calculan la infactibilidad total, los centroides y la distancia total intra-cluster de la partición en [Algoritmo 5](#), [Algoritmo 6](#) y [Algoritmo 7](#) respectivamente.

Algorithm 3: distanciaEuclidea

Input: vector u , vector v , con $\mathbf{tam}(u) == \mathbf{tam}(v)$

```

1 dist := 0.0;
2 for  $i := 0$  hasta  $\mathbf{tam}(u)$  do
3   | dist +=  $(u[i] - v[i])**2$ 
4 end
5 return sqrt(dist);
```

Algorithm 4: distanciaManhattan

Input: vector u , vector v , con $\mathbf{tam}(u) == \mathbf{tam}(v)$

```

1 dist := 0.0;
2 for  $i := 0$  hasta  $\mathbf{tam}(u)$  do
3   | dist +=  $|u[i] - v[i]|$ 
4 end
5 return dist;
```

Algorithm 5: calcularInfactibilidad

Input: vector solución s , vector de restricciones ML, vector de restricciones CL

```

1 infactibilidad := 0;
2 for  $i := 0$  hasta  $\mathbf{tam}(ML)$  do
3   | if  $s[ML[i,0]] \neq s[ML[i,1]]$  then
4     | infactibilidad++;
5   | end
6 end
7 for  $i := 0$  hasta  $\mathbf{tam}(CL)$  do
8   | if  $s[CL[i,0]] == s[CL[i,1]]$  then
9     | infactibilidad++;
10  | end
11 end
12 return infactibilidad;
```

4. Problema del agrupamiento con restricciones

Algorithm 6: calcularCentroides

Input: conjunto de datos X , vector solucion s , número de agrupamientos k

```
1 tamaños := vector( $k$ , 0);
2 centroides := matriz(( $k$ , tam( $X[0]$ ), 0.0);
3 for  $i := 0$  hasta tam( $X$ ) do
4   for  $j := 0$  hasta tam( $X[0]$ ) do
5     centroides[ $s[i]$ ][ $j$ ] +=  $X[i, j]$ ;
6     tamaños[ $s[i]$ ]++;
7   end
8 end
9 for  $i := 0$  hasta  $k$  do
10   if tamaños[ $i$ ] > 0 then
11     for  $j := 0$  hasta tam( $X[0]$ ) do
12       centroides[ $i$ ][ $j$ ] /= tamaños[ $i$ ];
13     end
14   end
15 end
16 return centroides;
```

Algorithm 7: calcularDistanciaIC

Input: conjunto de datos X , vector solucion s , vector de centroides $centr$

```
1 tamaños := vector(tam( $centr$ ), 0);
2 distancias := vector(tam( $centr$ ), 0.0);
3 distIC := 0.0
4 for  $i := 0$  hasta tam( $X$ ) do
5   distancias[ $s[i]$ ] += distanciaEuclidea( $X[i]$ ,  $centr[s[i]]$ );
6   tamaños[ $s[i]$ ]++;
7 end
8 for  $i := 0$  hasta tam( $centr$ ) do
9   distancias[ $i$ ] /= tamaños[ $i$ ];
10  distIC += distancias[ $i$ ];
11 end
12 return distIC;
```

5. Adaptación de los mapas autoorganizativos al clustering

En primer lugar para poder usar los mapas auto-organizativos para el problema de clustering con restricciones, vamos a necesitar adaptar su salida a la resolución de un problema de clustering común, en el que el número de clusters de salida no tiene por qué coincidir con el número de neuronas del mapa.

Cuando el número de clusters deseado es diferente del número de neuronas del SOM, se requieren pasos adicionales para analizar y agrupar los puntos del mapa de salida en el número deseado de clusters. Por tanto, se necesita adaptar la salida del SOM para poder ajustarse al problema de clustering.

Para no alterar el algoritmo original del proceso de autoorganización, voy a ampliar el modelo añadiendo un proceso de agrupación independiente que toma la salida generada por la red SOM para llegar al número deseado de clusters. Este proceso automático que voy a utilizar y que complementa el uso de las redes SOM de Kohonen, fue propuesto en [Mur95] y se trata de un método de agrupación aglomerativa con restricciones de contigüidad.

5.1. Método de agrupación por contigüidad

El método agrupa la salida del SOM basándose en un criterio de desviación cuadrática mínima para fusionar los nodos vecinos. El razonamiento es que las redes SOM mantendrán las relaciones topológicas originales; por lo tanto, los nodos que están estrechamente ubicados en el mapa deben tener centros de cluster similares.

Con este fundamento, se va a implementar un método que consiste en un algoritmo de agrupación con restricciones de contigüidad basado en un criterio de desviación cuadrática mínima. El criterio fue propuesto en [Kiao1] a partir de modificar el criterio de [Mur95] de distancia mínima. Trata de minimizar la desviación cuadrática global dentro del cluster en cada paso del proceso.

El método comienza con cada nodo del mapa representando un grupo, y para todos estos grupos se calcula su centroide. A continuación, intentamos fusionar dos grupos vecinos para que el resultado de la fusión mantenga la desviación cuadrática mínima global para ese número de conglomerados. El proceso de fusión se repite hasta que se haya obtenido un número de clusters especificado por el usuario o cuando sólo quede un cluster. El proceso detallado se describe a continuación:

5. Adaptación de los mapas autoorganizativos al clustering

1. Para cada partición C_i asociada al nodo y_i , se calcula su centroide μ_i como

$$\mu_i = \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j$$

donde $|C_i|$ es el número de instancias de la partición C_i .

2. Crear un cluster G_k para cada y_i si $|C_i| > 0$.
3. Calcule la desviación cuadrática global del mapa:
 - a) Sumar el cuadrado de la distancia entre el vector de entrada x y el centroide del cluster $\mu_k \forall x \in G_k$. Entonces, para cada cluster G_k

$$V_k = \sum ||x - \mu_k||^2, \text{ con } x \in G_k$$

- b) Desviación cuadrática total de todos los grupos. La desviación cuadrática global del mapa se define como:

$$V_{Total} = \sum V_k$$

4. Para cada par de grupos vecinos se calcula la desviación cuadrática total del mapa si los dos grupos se fusionaran. Y entre todas las posibles fusiones, se realiza la que que minimice la desviación cuadrática global.

- a) Se calcula el nuevo centroide para $G_{p,q}$ si G_p y G_q fueran agrupadas:

$$G_{pq} = \frac{|G_p|G_p + |G_q|G_q}{|G_p| + |G_q|}$$

- b) Se calcula la nueva desviación cuadrática si G_p y G_q fueran fusionadas

$$V_{pq} = \sum ||x - \mu_i||^2 \forall x \in G_p \cup G_q$$

- c) Se calcula la nueva desviación cuadrática global fusionando G_p y G_q

$$V_{pq-Total} = V_{Total} + V_{pq} - V_p - V_q$$

- d) Calcula el $V_{pq-Total}$ para cada par p y q en el mapa. Para cada iteración, los grupos p y q deben estar dentro de una distancia radial fija en la cuadrícula. Empezamos con una distancia radial = 1, por lo que para cada nodo hay cuatro nodos vecinos dentro de esa distancia (usando la distancia de Manhattan).

Aumentamos la distancia radial en uno cada vez si no hay ningún grupo vecino dentro de la distancia radial actual para todos los grupos k . Finalmente, fusionamos los dos grupos que resultan en la desviación cuadrática mínima global.

- e) Se modifican los índices de acuerdo al cambio en la lista de grupos y se actualiza tanto V_{Total} como el centroide del nuevo grupo obtenido de los dos nuevos fusionados.

5. Repita el paso 4 hasta alcanzar un solo cluster o el número de clusters preestablecido.

Parte III.

Propuestas de algoritmos con mapas autoorganizativos para agrupamiento con restricciones

6. SCCSOM

Este algoritmo es mi propuesta para la resolución del problema de clustering con restricciones “soft” utilizando un mapa auto-organizativo. He denominado al algoritmo “Soft Constrained Clustering SOM”, pero me referiré a él con el nombre SCCSOM.

6.1. Definición y explicación del SCCSOM

Sea $x \in X$ el conjunto de instancias para clasificar y sea Y el conjunto de neuronas del mapa. El algoritmo para cada entrada $x \in X$, tendrá como estrategia elegir la neurona y_i que haga que al asignarse x a su cluster C_i se produzca menos infactibilidad total, es decir, se elige la neurona y_i de forma que se minimice el número de restricciones incumplidas. En caso de que dos neuronas empaten con este criterio de elección, se elegirá la neurona con valor más cercano a x según la distancia euclídea. Con este criterio se busca satisfacer en primer lugar el máximo número de restricciones con el paso de las épocas y en segundo lugar, minimizar la desviación general de la partición resultante.

Para llevar a cabo este método es necesario que durante el entrenamiento del mapa, para cada entrada $x \in X$, se tenga la asignación de las instancias a los clusters para así poder calcular la infactibilidad total y se pueda obtener la mejor neurona para una entrada x . Esto se consigue al ir obteniendo la mejor neurona (BMU) durante el entrenamiento de cada $x \in X$ en cada época y anotar que la instancia pertenece al cluster asociado a su neurona ganadora.

Por este motivo de tener que calcular la infactibilidad no se debe usar el SOM básico ya que no tendría sentido obtener las neuronas ganadoras para todas las instancias después de cada época, pues sólo se ha entrenado con una sola instancia. Por tanto, se necesita usar la variante del SOM expuesta en la [Algoritmo 2](#), que permite en un número menor de épocas que las neuronas converjan al entrenarse con todo el conjunto en cada época. De esta forma se facilita la asignación de las instancias a los clusters durante el entrenamiento y además se tiene la certeza de que todas las instancias han sido asignadas a un cluster en el mismo entrenamiento cuando el número de épocas sea mayor que 1.

Como detalles finales se usa la función Gaussiana como función de vecindad y antes de cada época se barajará el vector de entradas para que el orden de asignación no sea siempre el mismo. Entonces al ejecutar el algoritmo durante varias épocas el orden en cada una de ellas no influye demasiado en el proceso.

6.2. Explicación de métodos

A continuación voy a explicar los métodos y funciones que aparecen en la descripción del pseudo-código del algoritmo [8](#).

6. SCCSOM

- **Clase SOM.** Clase que engloba el comportamiento del mapa SOM manejando la estructura de datos del conjunto de neuronas. Tiene los siguientes métodos de clase:
 - **bestNeuronSoft.** Dado un vector de entrada x , devuelve la neurona con cuya asignación a su cluster se minimice el número de restricciones totales incumplidas. En caso de que exista más de una con este criterio, se devuelve la más cercana a x según la distancia euclídea.
 - **cluster.** Devuelve y/o modifica el cluster al que se asocia una entrada x .
 - **adjust.** Dada una entrada x y una neurona y , ajusta el vecindario de la neurona y a la entrada x .
 - **updateParams.** Actualiza los parámetros, α y σ , que se modifican según el paso de las épocas.
 - **generateKClusters.** Dado un número de clusters k menor o igual que el número de neuronas, agrupa los clusters generados por las neuronas en k clusters según el criterio de [Sección 5.1](#).
 - **solution.** Devuelve un vector con la asignación de cada instancia a un cluster.
- **randInit.** Crea un mapa SOM de dimensión m , almacena un factor de aprendizaje inicial e inicializa un conjunto de neuronas que tengan la dimensión de los vectores de entrada.
- **randomShuffle.** Baraja aleatoriamente un conjunto o vector.

Algorithm 8: SCCSOM

Input : conjunto de datos X , vector de restricciones ML , vector de restricciones CL , dimensión del mapa m , número de clusters k , número de épocas n

```

1 # Se inicializan aleatoriamente las neuronas del mapa
2 # y se establece un factor de aprendizaje inicial  $\alpha$ .
3 som := randInit( $m, \alpha$ );
4 # Creo un vector de índices de  $X$ , para manejar las instancias.
5 indices :=  $\{0, \dots, \text{tam}(X)\}$ ;
6 for  $i := 0$  hasta  $n$  do
7   # Se reordenan los índices aleatoriamente.
8   indices := randomShuffle(indices);
9   # Entrenamos el mapa con todas las instancias de  $X$ .
10  for  $j := 0$  hasta  $\text{tam}(X)$  do
11     $x := X[\text{indices}[j]]$ ;
12    # Se selecciona la neurona más cercana a  $x$  que minimiza la infactibilidad total.
13     $y := \text{som.bestNeuronSoft}(x)$ ;
14    # Se asigna la instancia al cluster de la neurona.
15     $\text{som.cluster}(x) := y$ ;
16    # Finalmente se ajusta la neurona ganadora y su vecindario al vector de entrada.
17     $\text{som.adjust}(y, x)$ ;
18    # Se actualiza  $\alpha$  y se disminuye el radio del vecindario.
19     $\text{som.updateParams}()$ ;
20  end
21 end
22 # Se generan los clusters por la cercanía a las neuronas, y se agrupan hasta obtener sólo  $k$ .
23  $\text{som.generateKClusters}(k)$ ;
24 # Se devuelve un vector con la asignación de cada instancia a un cluster.
25 return  $\text{som.solution}()$ ;
```

7. CS2S

Este algoritmo se basa en el algoritmo CS2GS de [AYH15], que utiliza como mapa auto-organizativo un GSOM, que es una variante creciente del SOM. De forma más específica, GSOM es un mapa SOM que parte de un número inicial de neuronas y durante el entrenamiento puede crear neuronas nuevas, aumentando el tamaño del mapa.

7.1. CS2GS

Una de las características principales de CS2GS es que en cada época el vector de entrada no es una instancia del conjunto de datos a clasificar, si no que es una tripleta (x_i, x_j, b) con $b \in \{1, -1, 0\}$ según la restricción que exista sobre x_i y x_j . La idea principal del algoritmo es adaptar el mapa con x_i y x_j , eligiendo para cada uno su neurona más cercana y adaptando su vecindario, salvo en el caso de que aparezca como entrada una restricción ML o CL. En caso de que exista restricción entre x_i y x_j , la elección de las mejores neuronas para cada uno también está condicionada por la restricción.

En el caso de una restricción ML, se buscan dos neuronas cercanas a x_i y x_j , y_i e y_j respectivamente, que además entre ellas cumplan la siguiente restricción en su distancia:

$$d(y_i, y_j) < \epsilon,$$

con un valor de ϵ elegido como parámetro.

En cambio, en el caso de una restricción CL, se buscan dos neuronas cercanas a x_i y x_j , y_i e y_j respectivamente, que además entre ellas cumplan la siguiente restricción en su distancia:

$$d(y_i, y_j) > \rho,$$

con un valor de ρ elegido como parámetro.

Otra característica fundamental del algoritmo es que no almacena el cluster de las instancias durante el entrenamiento, sino que tan sólo adapta la red con la instancia. Y una vez que la red ha sido entrenada, se obtiene el cluster que se le asigna a cada instancia simplemente agrupando a la neurona más cercana sin tener en cuenta las restricciones, que ya habrán tenido su influencia durante el entrenamiento.

Esta idea se lleva a cabo en dos capas. Los datos de entrada fluyen a través de estas capas y, en última instancia, se asignan al cluster apropiado. Un esquema abstracto de este procedimiento se muestra en la [Figura 7.1](#). La red corrige dinámicamente la violación de las restricciones adaptando una matriz de pesos entre capas y haciendo crecer el número de nodos (neuronas) en la primera capa del GSOM.

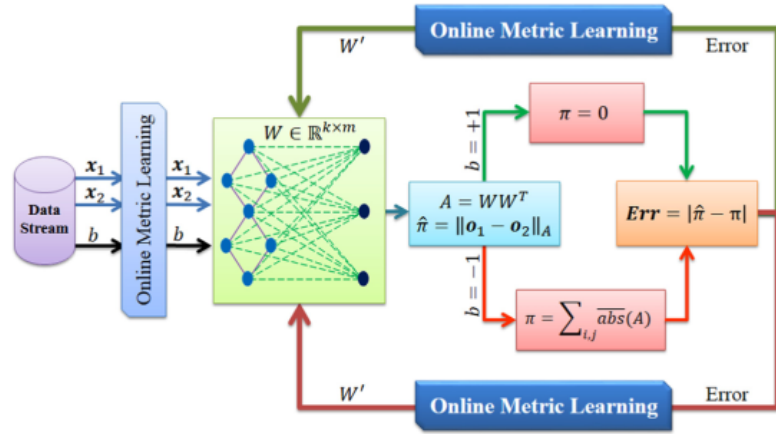


Figura 7.1.: Esquema del algoritmo CS2GS.

7.2. Adaptación a CS2S

El algoritmo CS2S tan sólo toma la idea general de CS2GS, ya que CS2S es un algoritmo muy complejo que además de usar un GSOM, no usa la distancia euclídea sino una distancia cuadrática distinta, A , que se va adaptando con las épocas y que usa una matriz de pesos, W , que se va modificando con propagación hacia atrás, como se ve en el diseño de la [Figura 7.1](#).

El algoritmo CS2S usará como SOM la variante explicada en [Algoritmo 2](#) en vez del GSOM, ya que la visión y explicación de GSOM se escapa a lo abordado en este trabajo. Además, por la misma razón se utiliza también la distancia euclídea en vez de la cuadrática. Con estas modificaciones se podrá comparar lo más justamente posible los algoritmos SOM para Clustering con Restricciones que se exponen en el trabajo. Del CS2GS me quedará con la idea de tomar como entrada tripletas (x_i, x_j, b) que son simplemente pares de instancias con su restricción. Definiendo este conjunto de entrada como Z se tiene que:

$$Z = \{(x_i, x_j, b) \mid \forall x_i, x_j \in X \text{ con } i \neq j, \text{ y } b \in \{0, 1, -1\}\}$$

Y como

$$\bigcup_{i \neq j} \{(x_i, x_j, 1)\} = ML \text{ y } \bigcup_{i \neq j} \{(x_i, x_j, -1)\} = CL,$$

podemos definir un nuevo conjunto compuesto por todos los pares de instancias que no tienen asociada una restricción, WL, con

$$\bigcup_{i \neq j} \{(x_i, x_j, 0)\} = WL,$$

y con estos tres conjuntos se tiene que:

$$Z = ML \cup CL \cup WL.$$

Por tanto, las entradas de CS2S van a ser las restricciones y los pares sin restricción del conjunto WL.

La estrategia del algoritmo será la misma que CS2GS, en cada época se entrenará primero el mapa con los pares de instancias sin restricción, y luego los pares con las restricciones, buscando separar o agrupar las instancias. Según el tipo de restricción se elegirán las dos neuronas ganadoras más cercanas pero que entre ellas estén a una cierta distancia.

Por último, como se ha dicho anteriormente, en vez del SOM original, en este algoritmo se utiliza una variante del SOM, explicada en [Algoritmo 2](#), que entrenará al SOM con el conjunto completo de tripletas.

7.3. Explicación de métodos

A continuación voy a explicar los métodos y funciones que aparecen en la descripción en pseudo-código del [Algoritmo 9](#).

- **Clase SOM.** Clase que engloba el comportamiento del mapa SOM. Maneja la estructura de datos del conjunto de neuronas. Tiene los siguientes métodos de clase:
 - **bestNeuronDistance.** Dado un vector de entrada x , devuelve la neurona más cercana a x según la distancia euclídea.
 - **bestNeuronDistanceML.** Dado un vector de entrada x , una neurona y_E y un escalar ϵ ; el método devuelve la neurona y más cercana a x cuya distancia $d(y, y_E) < \epsilon$. La distancia que se utiliza es la distancia euclídea.
 - **bestNeuronDistanceCL.** Dado un vector de entrada x , una neurona y_E y un escalar ρ ; el método devuelve la neurona y más cercana a x cuya distancia $d(y, y_E) > \rho$. La distancia que se utiliza es la distancia euclídea.
 - **clustering.** Asigna a cada instancia su cluster más cercano, es decir el asociado a su neurona más cercana.
 - **adjust.** Dada una entrada x y una neurona y , ajusta el vecindario de la neurona y a la entrada x .
 - **updateParams.** Actualiza los parámetros, α y σ , que se modifican según el paso de las épocas.
 - **generateKClusters.** Dado un número de clusters k menor o igual que el número de neuronas, agrupa los clusters generados por las neuronas en k clusters según el criterio de la [Sección 5.1](#).
 - **solution.** Devuelve un vector con la asignación de cada instancia a un cluster.
- **randInit.** Crea un mapa SOM de dimensión m , almacena un factor de aprendizaje inicial e inicializa un conjunto de neuronas que tengan la dimensión de los vectores de entrada.
- **randomShuffle.** Baraja aleatoriamente un conjunto o vector.

Algorithm 9: CS2S

Input : conjunto de datos X , vector de restricciones ML , vector de restricciones CL , vector sin restricciones WL , dimensión del mapa m , número de clusters k , número de épocas n , distancia ML ϵ , distancia CL ρ

```

1  # Se inicializan aleatoriamente las neuronas del mapa
2  # y se establece un factor de aprendizaje inicial  $\alpha$ .
3  som := randInit(m,  $\alpha$ );
4  for  $i := 0$  hasta  $n$  do
5      # Se barajan  $ML$ ,  $CL$  y  $WL$ .
6       $ML := \text{randomShuffle}(ML)$ ;
7       $CL := \text{randomShuffle}(CL)$ ;
8       $WL := \text{randomShuffle}(WL)$ ;
9      # Entrenamos el mapa con todas las instancias de  $ML$ ,  $CL$  y  $WL$ .
10     for  $j := 0$  hasta  $\text{tam}(WL)$  do
11          $x_1 := WL[j,0]$ ,  $x_2 := WL[j,1]$ ;
12         # Se seleccionan las neuronas más cercanas a  $x_1$  y  $x_2$ .
13          $y_1 := \text{som.bestNeuronDistance}(x_1)$ ;
14          $y_2 := \text{som.bestNeuronDistance}(x_2)$ ;
15         # Se ajustan las neuronas ganadoras y sus vecindarios a los vectores de entrada.
16          $\text{som.adjust}(y_1, x_1)$ ,  $\text{som.adjust}(y_2, x_2)$ ;
17     end
18     for  $j := 0$  hasta  $\text{tam}(ML)$  do
19          $x_1 := WL[j,0]$ ,  $x_2 := WL[j,1]$ ;
20         # Se seleccionan las neuronas más cercanas a  $x_1$  y  $x_2$ , cumpliendo que  $d(y_1, y_2) < \epsilon$ .
21          $y_1 := \text{som.bestNeuronDistance}(x_1)$ ;
22          $y_2 := \text{som.bestNeuronDistanceML}(x_2, y_1, \epsilon)$ ;
23         # Se ajustan las neuronas ganadoras y sus vecindarios a los vectores de entrada.
24          $\text{som.adjust}(y_1, x_1)$ ,  $\text{som.adjust}(y_2, x_2)$ ;
25     end
26     for  $j := 0$  hasta  $\text{tam}(CL)$  do
27          $x_1 := WL[j,0]$ ,  $x_2 := WL[j,1]$ ;
28         # Se seleccionan las neuronas más cercanas a  $x_1$  y  $x_2$ , cumpliendo que  $d(y_1, y_2) > \rho$ .
29          $y_1 := \text{som.bestNeuronDistance}(x_1)$ ;
30          $y_2 := \text{som.bestNeuronDistanceCL}(x_2, y_1, \rho)$ ;
31         # Se ajustan las neuronas ganadoras y sus vecindarios a los vectores de entrada.
32          $\text{som.adjust}(y_1, x_1)$ ,  $\text{som.adjust}(y_2, x_2)$ ;
33     end
34     # Se asignan finalmente las instancias al cluster de la neurona.
35      $\text{som.clustering}(x_1) := y_1$ ,  $\text{som.cluster}(x_2) := y_2$ ;
36     # Se actualiza  $\alpha$  y se disminuye el radio del vecindario.
37      $\text{som.updateParams}()$ ;
38 end
39 # Se generan los clusters por la cercanía a las neuronas, y se agrupan hasta obtener sólo  $k$ .
40  $\text{som.generateKClusters}(k)$ ;
41 # Se devuelve un vector con la asignación de cada instancia a un cluster.
42 return  $\text{som.solution}()$ ;

```

8. CopSoft-SOM

Este algoritmo se basa en el algoritmo Cop-SOM de [YTLR12], que utiliza un mapa auto-organizativo para resolver el problema de clustering con restricciones “hard”. Consiste en una adaptación de Cop-SOM para manejar restricciones “soft” en vez de “hard” y usará la variante del SOM explicada en [Algoritmo 2](#).

8.1. Cop-SOM

En Cop-SOM se entrena el mapa eligiendo aleatoriamente un vector de entrada $x \in X$, y el entrenamiento se caracteriza porque la neurona ganadora es aquella más cercana a x que no viola ninguna restricción. En el caso de no existir ninguna neurona posible, llama a una función denominada “Process-violation” indicando que cualquier asignación de x implicaría la violación de una restricción.

La finalidad de la llamada a “Process-violation” es reasignar algunas instancias, entendiendo que el orden de asignación de estas puede llevar a que se cometa la violación forzosa de alguna restricción, debido a las asignaciones hechas anteriormente.

A partir de la descripción del algoritmo y de los experimentos, el orden de asignación puede llevar a obtener un resultado no válido, lo que se puede ilustrar con la [Figura 8.1](#). Se tienen $(x_i, x_k), (x_j, x_k) \in CL$, con x_i y x_j que ya han sido asignados a los clusters C_a y C_b respectivamente antes de decidir el cluster de x_k . Cuando llega el momento de asignar x_k , no hay ningún cluster correcto que satisfaga todas las restricciones. En cambio con otro orden de asignación, no se incumpliría ninguna restricción, puesto que x_i y x_j estarían en el mismo cluster, y x_k en el otro.

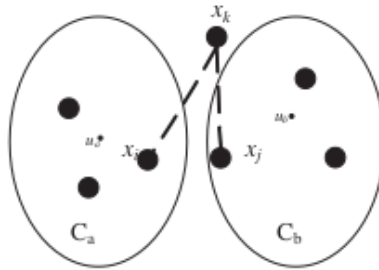


Figura 8.1.: Error de asignación de x_k en Cop-SOM.

Con este sencillo ejemplo, se puede observar como un orden de asignación inadecuado hace que x_i y x_j estén separados e indirectamente conduce a la violación de una restricción “cannot-link” al intentar asignar x_k .

8. CopSoft-SOM

Con la llamada a “Process-violation” se busca resolver este problema. El procedimiento es el siguiente, se llama con una instancia x_i que incumple alguna restricción sea cual sea la neurona que se le asigne. Entonces dentro de la función se asigna $x : i$ a la neurona más cercana sin tener en cuenta las restricciones incumplidas. Estas restricciones incumplidas que son de la forma (x_i, x_j) o (x_j, x_i) , se listan en un conjunto I , y se obtiene un conjunto S tal que:

$$S = \{x_j : (x_i, x_j) \in I \vee (x_j, x_i) \in I\}$$

Finalmente, para cumplir las restricciones de I , ahora incumplidas por la asignación de x_i , se reasignarán todas las instancias de S , buscando para cada x_j su mejor neurona, es decir, aquella más cercana sin violar ninguna restricción. Cuando una instancia de S se asigna se elimina de S . “Process-violation” acaba cuando se vacía el vector S . En el caso de que al asignar una instancia x_j de S , se incumpla alguna restricción sea cual sea la neurona que se le asigne, se llama también a “Process-violation”, ahora con la instancia x_s . A causa de esto, Process-violation es una función recursiva, dando lugar a la recursión sólo si las instancias que necesitan ser reasignadas también sufren una asignación fallida. Esta situación existe, pero no se da con frecuencia cuando el número de restricciones es pequeño.

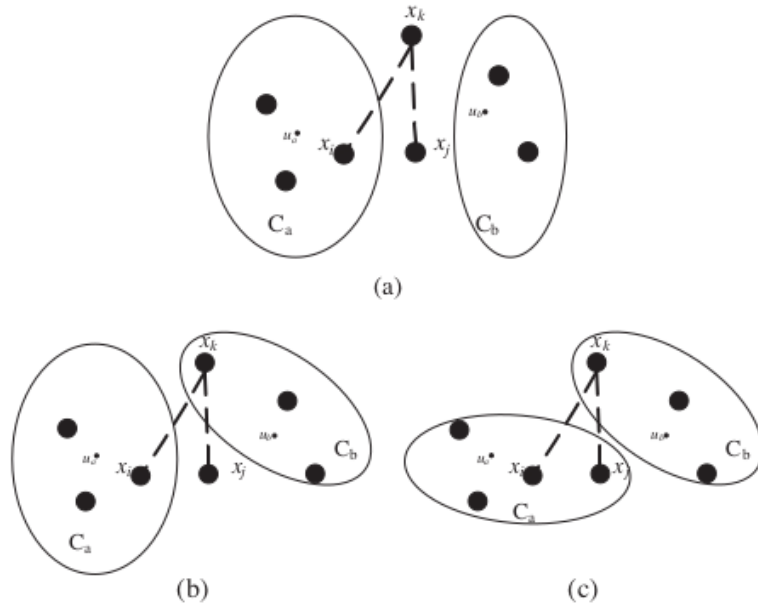


Figura 8.2.: Proceso de asignación de conflictos en Cop-SOM.

8.2. Adaptación a Cop-Soft SOM

La adaptación más significativa se va a producir en el proceso de asignación de conflictos. Se va a sustituir la función recursiva “Process-violation” que no permite que se incumpla ninguna restricción, por una nueva función denominada “ProcessViolationSoft”.

En esta nueva función, como pasaba en “Process-violation”, se asigna la instancia x conflictiva con la que se llama a “ProcessViolationSoft” al cluster definido por la neurona más cercana, generándose también el conjunto S . El cambio se produce en que para cada elemento que haya en S , x_s , si durante su asignación no se encuentra ningún cluster porque incumpla alguna restricción, en vez de llamar de nuevo a la función “ProcessViolationSoft” recursivamente, esta instancia x_s se asigna el cluster con el que se incumplan las mínimas restricciones posibles como sucedía en la asignación del SCCSOM (Capítulo 6). De esta forma el algoritmo da lugar a una asignación que incumpla restricciones y por tanto el manejo de restricciones sea “soft”.

Esta adaptación se justifica en que la base de Cop-SOM es el orden, y por tanto si se va a violar una restricción, reasignando unas pocas instancias se soluciona. Además COP-SOM se fundamenta en que con pocas restricciones la llamada a Process-Violation no suele ser recursiva porque todos los elementos de S suelen asignarse de nuevo sin violar ninguna restricción. Por tanto, con esta adaptación, se consigue cumplir esa idea, y además, con tal de que el proceso no entre en una recursión se permite incumplir restricciones al asignar elementos de S en “ProcessViolationSoft”, teniendo en cuenta que este número de restricciones incumplidas decrecerá con el paso de las épocas.

Por último, en vez del SOM original, en este algoritmo se utilizará una variante del SOM, explicada en Algoritmo 2, y que consiste en que en cada época se entrenará al SOM con el conjunto completo de datos en vez de con sólo una instancia. Y además en la inicialización del mapa SOM con los datos, añade inicialmente las instancias a un cluster de neurona, iniciando el entrenamiento a partir de un agrupamiento aleatorio.

8.3. Explicación de métodos

A continuación voy a explicar los métodos y funciones que aparecen en la descripción en pseudo-código del Algoritmo 10.

- **Clase SOM.** Clase que engloba el comportamiento del mapa SOM. Maneja la estructura de datos del conjunto de neuronas. Tiene los siguientes métodos de clase:
 - **ProcessViolationSoft.** Dada una entrada x cuya asignación provoca la violación de una restricción, activa un procedimiento con la intención de que se asigne sin violar ninguna. Este método se ha detallado en el Algoritmo 11.
 - **bestNeuronHard.** Dado un vector de entrada x , devuelve la neurona con cuya asignación a su cluster no se viole ninguna restricción. En caso de que exista con este criterio más de una, se devuelve la más cercana a x según la distancia euclídea, y si no existe ninguna posible, se devuelve -1 .
 - **bestNeuronSoft.** Dado un vector de entrada x , devuelve la neurona con cuya asignación a su cluster se minimice el número de restricciones totales incum-

8. CopSoft-SOM

plidas. En caso de que exista más de una con este criterio, se devuelve la más cercana a x según la distancia euclídea.

- **bestNeuronDistance.** Dado un vector de entrada x , devuelve la neurona más cercana a x según la distancia euclídea.
 - **cluster.** Devuelve y/o modifica el cluster al que se asocia una entrada x .
 - **adjust.** Dada una entrada x y una neurona y , ajusta el vecindario de la neurona y a la entrada x .
 - **updateParams.** Actualiza los parámetros, α y σ , que se modifican según el paso de las épocas.
 - **generateKClusters.** Dado un número de clusters k menor o igual que el número de neuronas, agrupa los clusters generados por las neuronas en k clusters según el criterio de [Sección 5.1](#).
 - **solution.** Devuelve un vector con la asignación de cada instancia a un cluster.
- **randInit.** Crea un mapa SOM de dimensión m , almacena un factor de aprendizaje inicial e inicializa un conjunto de neuronas que tengan la dimensión de los vectores de entrada.
 - **randomShuffle.** Baraja aleatoriamente un conjunto o vector.

Algorithm 10: CopSoft-SOM

Input : conjunto de datos X , vector de restricciones ML , vector de restricciones CL , dimensión del mapa m , número de clusters k , número de épocas n

```

1 # Se inicializan aleatoriamente las neuronas del mapa
2 # y se establece un factor de aprendizaje inicial  $\alpha$ .
3 som := randInit( $m, \alpha$ );
4 # Creo un vector de índices de  $X$ , para manejar las instancias.
5 indices :=  $\{0, \dots, \text{tam}(X)\}$ ;
6 for  $i := 0$  hasta  $n$  do
7   # Se reordenan los índices aleatoriamente.
8   indices := randomShuffle(indices);
9   # Entrenamos el mapa con todas las instancias de  $X$ .
10  for  $j := 0$  hasta  $\text{tam}(X)$  do
11     $x := X[\text{indices}[j]]$ ;
12    # Se selecciona la neurona más cercana a  $x$  que no incumple ninguna restricción.
13     $y := \text{som.bestNeuronHard}(x)$ ;
14    # Si no se ha encontrado ninguna porque en todas se incumple alguna restricción.
15    if  $y == -1$  then
16       $y := \text{som.ProcessViolationSoft}(x)$ ;
17    end
18    # Se asigna la instancia al cluster de la neurona.
19    som.cluster( $x$ ) :=  $y$ ;
20    # Finalmente se ajusta la neurona ganadora y su vecindario al vector de entrada.
21    som.adjust( $y, x$ );
22    # Se actualiza  $\alpha$  y se disminuye el radio del vecindario.
23    som.updateParams();
24  end
25 end
26 # Se generan los clusters por la cercanía a las neuronas, y se agrupan hasta obtener sólo  $k$ .
27 som.generateKClusters( $k$ );
28 # Se devuelve un vector con la asignación de cada instancia a un cluster.
29 return som.solution();

```

8. CopSoft-SOM

Algorithm 11: ProcessViolationSoft

Input : vector de entrada x , conjunto de datos X , vector de restricciones ML , vector de restricciones CL , mapa som

```

1 # Se selecciona la neurona más cercana a  $x$ .
2  $y := som.bestNeuronDistance(x)$ ;
3  $S := []$ ;
4 # Recorro el conjunto de restricciones Must-Link.
5 for  $i := 0$  hasta  $tam(ML)$  do
6     # Se seleccionan las restricciones en las que aparezca  $x$ .
7     if  $((x,s) == ML[i] \text{ or } (s,x) == ML[i])$  and  $som.cluster(s) \neq y$  then
8         # Los  $s$  que tengan una restricción  $ML$  y no estén en  $y$  se incluyen en  $S$ .
9          $S.add(s)$ ;
10    end
11 end
12 # Recorro el conjunto de restricciones Cannot-Link.
13 for  $i := 0$  hasta  $tam(CL)$  do
14     # Se seleccionan las restricciones en las que aparezca  $x$ .
15     if  $((x,s) == CL[i] \text{ or } (s,x) == CL[i])$  and  $som.cluster(s) == y$  then
16         # Los  $s$  que tengan una restricción  $CL$  y estén en  $y$  se incluyen en  $S$ .
17          $S.add(s)$ ;
18    end
19 end
20 # Reasignamos las instancias incluidas en  $S$ .
21 for  $j := 0$  hasta  $tam(S)$  do
22      $t := som.bestNeuronSoft(S[j])$ ;
23     # Se asigna la instancia al cluster de la neurona.
24      $som.cluster(S[j]) := t$ ;
25 end
26 return  $y$ ;

```

9. Ensembles

Un “ensemble” es un conjunto de modelos de aprendizaje automático. Cada modelo produce una predicción diferente y luego estas se combinan para obtener una única predicción. La ventaja que obtenemos al combinar modelos diferentes es que como cada modelo funciona de forma diferente, sus errores tienden a compensarse. Esto resulta en un mejor error de generalización. De esta forma se consigue obtener un rendimiento predictivo mejor que el que podría obtenerse con cualquiera de los algoritmos de aprendizaje constituyentes por sí solos.

Inspirado en el “ensemble” de clasificadores múltiples, el método de “ensemble” para clustering ha demostrado mejorar el rendimiento de los algoritmos de clustering tradicionales. Este nuevo método integra múltiples soluciones al agrupamiento generadas por diferentes algoritmos, o por el mismo algoritmo pero con diferentes parámetros de inicialización. La agrupación final de consenso con su mayor estabilidad y robustez se obtiene después de tal combinación. El establecimiento de funciones de consenso es la cuestión clave para el conjunto de clustering.

En [FJ05] se exploró un enfoque de agrupación por acumulación de pruebas. Su enfoque convierte el conjunto de posibles particiones en una nueva medida de similitud entre patrones mediante la acumulación de coasociaciones de patrones por pares. Luego en [AJ09] se presentó un método semi-supervisado de clustering “ensemble” para combinar múltiples soluciones para clustering semi-supervisado y así poder asegurar el cumplimiento de las restricciones.

El conjunto de clustering en forma de matriz de coasociación obtiene un nuevo orden de asignación para las instancias de datos. Esta ordenación de las entradas es muy útil para muchos algoritmos de clustering semisupervisado, que son muy sensibles al orden de asignación de las instancias, como sucede en SSCSOM.

9.1. Un nuevo orden de asignación

9.1.1. La matriz de coasociación

Dado un conjunto de objetos de datos $X = \{x_1, x_2, \dots, x_N\}$ donde N es el número de datos y $x_i = \{x_{i1}, x_{i2}, \dots, x_{id}\}$, siendo d la dimensionalidad de las características. Supongamos que se generan múltiples particiones de X , es decir, soluciones a la agrupación, y las agrupamos en un conjunto Π :

$$\Pi = \{\pi_1, \pi_2, \dots, \pi_H\},$$

donde H es el número de particiones generadas.

Ahora definimos para cada $x_i \in X$ su correspondiente **vector de clases**

$$x_i \rightarrow \{\pi_1(x_i), \pi_2(x_i), \dots, \pi_H(x_i)\},$$

9. Ensembles

en el que se representa para cada solución de agrupación π_j el cluster asignado a x_i .

Entonces se construye una matriz de coasociación de dimensión $N \times N$, donde cada elemento a_{ij} representa el número de particiones en las que las instancias x_i y x_j están situadas en un mismo clúster. La entrada en la matriz de coasociación también indica un tipo de similitud entre dos objetos, y se define como sigue:

$$coasociacion(x_i, x_j) = \frac{1}{H} \sum_{h=1}^H \delta(\pi_h(x_i), \pi_h(x_j)).$$

9.1.2. El grado de certeza

Sean dos instancias $x_i, x_j \in X$, su $coasociacion(x_i, x_j)$ es un valor real entre 0 y 1 que representa una relación de similitud de cada par (x_i, x_j) de ser asignados juntos.

Cuando $coasociacion(x_i, x_j)$ es igual o cercana a 1, implica que es muy probable que x_i y x_j sean clasificados en el mismo cluster. Y cuando $coasociacion(x_i, x_j)$ es igual o cercana a 0, implica que es muy probable que x_i y x_j sean clasificados en diferentes clusters. Sin embargo, si $coasociacion(x_i, x_j)$ tiene un valor igual o cercano a 0.5, es bastante inseguro afirmar que los objetos están colocados en el mismo cluster o no. Por lo tanto, es crucial describir claramente la certeza de un solo objeto en lugar de dos objetos. Aquí, establecemos una función de mapeo por la matriz de coasociación dada en un valor que representa la asociación con todos los objetos. La función de mapeo se define como sigue:

$$certeza(x_i) = \frac{1}{N-1} \sum_{j=1, j \neq i}^N F(coasociacion(x_i, x_j)),$$

con

$$F(x) = \begin{cases} 1 - 2x & \text{si } 0 \leq x \leq 0.5 \\ 2x - 1 & \text{si } 0.5 < x \leq 1 \end{cases}$$

Entonces se tiene que $0 \leq certeza(x_i) \leq 1$. Nótese que $F(x)$ es una función continua y monótona en los intervalos $[0, 0.5]$ y $[0.5, 1]$. Considerando los elementos de la matriz de coasociación como la variable independiente de $F(x)$, se tiene $F(Coasociacin(x_i, x_j))$ denota la certeza relativa entre x_i y x_j .

Cuando $coasociacion(x_i, x_j)$ es igual o cercana a 1 o 0, implica que es muy probable o poco probable que se clasifiquen en el mismo cluster y que x_i es muy seguro en relación con x_j . En este caso, $F(coasociacin(x_i, x_j))$ es igual o cercano a 1.

Cuando $coasociacion(x_i, x_j)$ es igual o cercana a 0.5, implica que la relación de cluster es insegura entre x_i y x_j y por tanto, x_i es incierta en relación con x_j . En este caso, $F(coasociacion(x_i, x_j))$ es igual o cercana a 0.

9.2. Cambio de espacio de características

Sea $X = \{x_1, x_2, \dots, x_N\}$ un conjunto de datos donde $x_i = \{x_{i1}, x_{i2}, \dots, x_{id}\} \in \mathbb{R}^d$. Cuando se generan H configuraciones de clustering de X , el conjunto de datos original X con

características de dimensión d puede transformarse en una nueva matriz A del espacio de características $N \times H$ que se muestra en la **Tabla 9.1**. Cada entrada en A representa el cluster $\pi_j(x_i)$ en vez de las características originales de la instancia x_i y las H configuraciones de clustering son consideradas como un nuevo espacio de características H -dimensional.

	π_1	π_2	π_3	\cdots	π_H
x_1	$\pi_1(x_1)$	$\pi_2(x_1)$	$\pi_3(x_1)$	\cdots	$\pi_H(x_1)$
x_2	$\pi_1(x_2)$	$\pi_2(x_2)$	$\pi_3(x_2)$	\cdots	$\pi_H(x_2)$
x_3	$\pi_1(x_3)$	$\pi_2(x_3)$	$\pi_3(x_3)$	\cdots	$\pi_H(x_3)$
\cdots	\cdots	\cdots	\cdots	\cdots	\cdots
x_N	$\pi_1(x_N)$	$\pi_2(x_N)$	$\pi_3(x_N)$	\cdots	$\pi_H(x_N)$

Tabla 9.1.: Nuevo espacio de características.

9.3. Algoritmos de Ensemble

Para poder explorar la solución con métodos de “ensemble” al clustering con restricciones usando mapas auto-organizativos, se van a definir dos métodos de ordenación. El primero fue propuesto para un mapa auto-organizativo, y el segundo fue propuesto para K-Medias pero ha sido adaptado por mí para el caso de utilizar un mapa auto-organizado.

Para implementar esta asignación de orden en la entrada del SOM se va a modificar el algoritmo SCCSOM visto en el **Capítulo 6**, de forma que no se baraje el vector de entradas en cada época, sino que se ordene antes del entrenamiento conforme al criterio deseado y no se altere ese orden con el transcurrir de las épocas. Además con tal de darle aún más importancia a las restricciones y al conjunto de soluciones de entrada, aplico una modificación antes del método. Esta modificación consiste en incluir un hiperparámetro en el modelo, s , que define el número de un conjunto de soluciones Π' , siendo Π' el conjunto con las s mejores soluciones de Π . El criterio que marca la bondad es el cumplimiento de restricciones.

La modificación concluye al tomar Π' como entrada en vez de Π . De esta forma se queda un conjunto de soluciones menor pero más preciso.

9.3.1. SCCSOM-E

Se basa en el algoritmo Cop-SOM-E de [YTLR12], que define un método de ensemble sobre el algoritmo Cop-SOM aparecido también en [YTLR12] y que maneja restricciones “hard”. En **Figura 9.1** se presenta una imagen descriptiva de la estrategia de este algoritmo.

Se tiene que cuanto mayor sea la $\text{certeza}(x_i)$, mayor será el grado de certeza de esa instancia. Por tanto, antes puede ordenarse.

Muchos algoritmos de clustering semi-supervisado como los que funcionan con SOM son sensibles al orden de asignación de los objetos de datos. Si un objeto con la certeza baja se asigna antes de un objeto con la certeza alta y una restricción ML o CL se conecta entre ellos, hay una mayor probabilidad de identificar sus etiquetas de clase de forma inexacta.

9. Ensembles

Por lo tanto, reordenamos todos los objetos en un orden descendente según la certeza de cada objeto, produciéndose así un nuevo orden de asignación. El algoritmo se definirá en pseudo-código con más detalle en [Algoritmo 12](#).

Algorithm 12: SCCSOM-E

Input : conjunto de datos X , soluciones al agrupamiento Π , vector de restricciones ML , vector de restricciones CL , dimensión del mapa m , número de clusters k , número de épocas n , número de soluciones s , siendo $s < h$, con h el número de soluciones de Π

```

1 # Se escogen las  $s$  mejores soluciones según el cumplimiento de restricciones.
2  $\Pi' := \text{bestSolutions}(\Pi, s)$ ;
3 # Se transforma el conjunto de datos  $X$  que tiene  $n$  datos en un nuevo conjunto de datos  $A$ ,
  con dimensión  $n \times s$  siendo  $s$  el número de soluciones.
4  $A := \text{getNewSpace}(X, \Pi')$ ;
5 # Se toma como entrada de un algoritmo SOM para clustering con restricciones
6 # con la modificación de orden y se genera el resultado final de la agrupación.
7  $\text{som} := \text{SCCSOM}(A, ML, CL, m, k, \text{order} = E)$ ;
8 return  $\text{som.solution}()$ ;

```

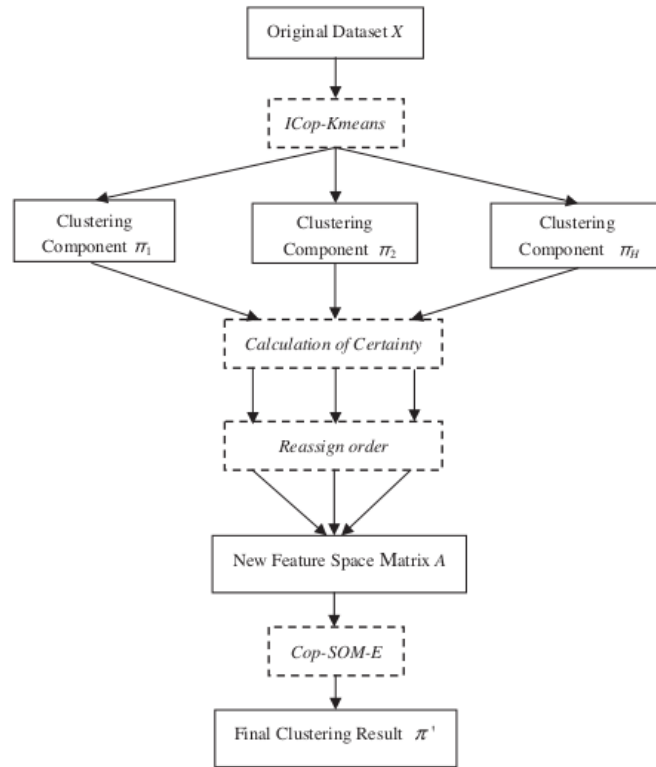


Figura 9.1.: Modelo de ejecución del algoritmo Cop-SOM-E con soluciones al clustering por K-medias.

9.3.2. SCCSOM-EAC-SL

Para este algoritmo se ha adaptado el “Constrained Evidence Accumulation Single-Link” (CC-EAC-SL) propuesto en [AJo9] y basado en K-Medias a SOM. Este algoritmo no hace uso de cálculo de la certeza sino que saca un nuevo orden de asignación directamente de la matriz de coasociación. En la ?? se muestra un esquema explicativo de la ejecución del algoritmo.

Este orden viene determinado de la siguiente manera: Se ordenan los elementos de la matriz de coasociación por orden descendente de forma que:

$$L = \{a_{ij} \in \text{coasociacion} : \text{con los } a_{ij} \text{ ordenados de forma descendente}\}$$

en una lista donde los valores están acotados, $a_{ij} \in [0, 1]$.

Luego se define un vector de instancias vacío S , y mientras que el número de instancias en S no sea igual que en el número de instancias en el conjunto de datos X , se va recorriendo el vector L de forma ordenada eligiendo cada vez un a_{ij} . Entonces se tiene para cada a_{ij} :

- Si $x_i \notin S$, $S.\text{add}(x_i)$.
- Si $x_j \notin S$, $S.\text{add}(x_j)$.

En donde el método **add**, lo que hace es añadir los elementos al final de la lista.

Con esta ordenación ya se puede definir la nueva versión del método para SOM, basta con modificar cualquiera de los algoritmos SOM vistos anteriormente (SCCSOM, SC2S o CopSoft-SOM) para no barajar el conjunto de entrada al SOM en cada época sino que se mantenga este orden definido en el vector que define el conjunto. El algoritmo se definirá en pseudo-código con más detalle en **Algoritmo 13**.

Algorithm 13: SCCSOM-EAC-SL

Input : conjunto de datos X , soluciones al agrupamiento Π , vector de restricciones ML , vector de restricciones CL , dimensión del mapa m , número de clusters k , número de épocas n , número de soluciones s , siendo $s < h$, con h el número de soluciones de Π

- 1 # Se escogen las s mejores soluciones según el cumplimiento de restricciones.
 - 2 $\Pi' := \text{bestSolutions}(\Pi, s)$;
 - 3 # Se transforma el conjunto de datos X que tiene n datos en un nuevo conjunto de datos A , con dimensión $n \times s$ siendo s el número de soluciones.
 - 4 $A := \text{getNewSpace}(X, \Pi')$;
 - 5 # Se toma como entrada de un algoritmo SOM para clustering con restricciones
 - 6 # con la modificación de orden y se genera el resultado final de la agrupación.
 - 7 $\text{som} := \text{SCCSOM}(A, ML, CL, m, k, n, \text{order} = \text{EAC-SL})$;
 - 8 **return** $\text{som.solution}()$;
-

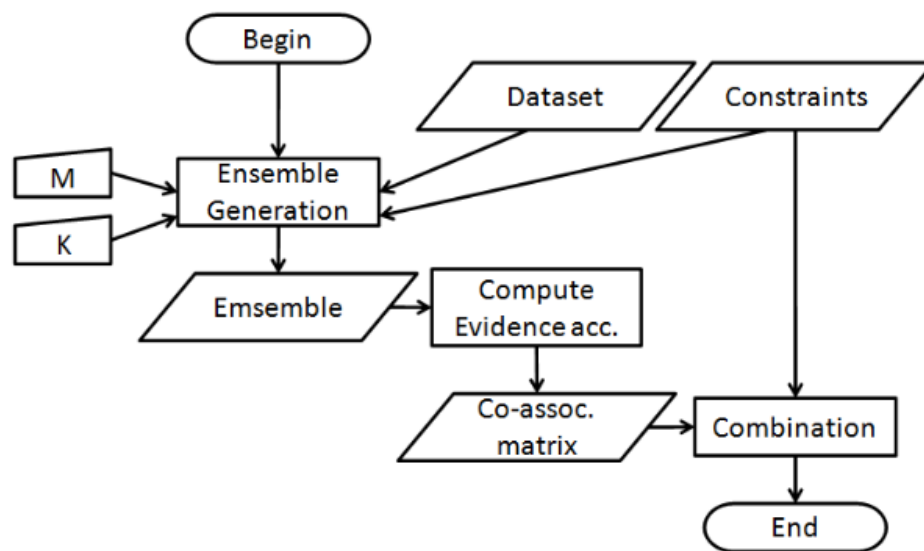


Figura 9.2.: Esquema de ejecución del CC-EAC-SL.

10. Implementación

Para implementar los algoritmos descritos en las secciones anteriores se ha realizado una librería en Python denominada SOMCC. Esta librería contiene tanto clases que corresponden a cada algoritmo como funciones de utilidad para el problema que se maneja. La estructura de clases es jerárquica y basada en herencia.

La librería se encuentra en [Python SOMCC](#).

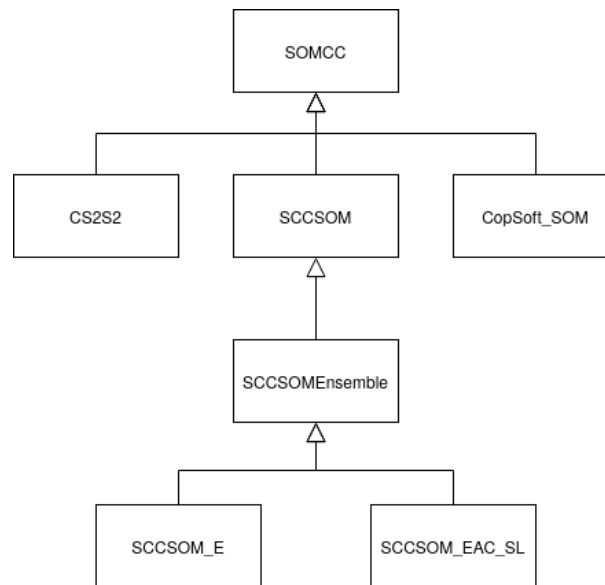


Figura 10.1.: Jerarquía de clases de SOMCC.

10.1. Documentación

A continuación voy a explicar por partes las clases, métodos y funciones definidos a modo de documentación para el usuario.

10.1.1. Funciones comunes

Obtiene la distancia euclídea entre los vectores a y b.
def distanceEu(a,b)

Obtiene la distancia de Manhattan entre los vectores a y b.
def distanceManh(a,b)

10. Implementación

```
# Dado un valor de distancia y un valor de desviacion estandar
# sigma, se calcula el valor de la funcion de vecindad usando
# la funcion gaussiana.
def neighborhoodFunction(dist, sigma)

# Devuelve una lista de restricciones a partir de una matriz de
# restricciones.
def constraintsList(restr)

# Devuelve si un cluster cl1 es vecino de otro, cl2, en el mapa
# a una determinada distancia de Manhattan dist.
def neighbour(cl1, cl2, dist)
```

10.1.1. Clase Cluster

```
# Clase que modula un Cluster de instancias.
class Cluster(object):

# Posee a __init__ como unico metodo.
# Se inicializa un cluster con su posicion en el mapa de neuronas,
# su centroide, su varianza y su lista de instancias.
def __init__(self, arrayPos, centroide, variance, inputs)
```

10.1.2. Métodos comunes a la clase SOM para clustering con restricciones

Se ha creado una clase SOM con la que es posible definir un mapa n-dimensional de neuronas (aunque nuestros experimentos han sido probados sólo con 2 y 3 dimensiones), la función de vecindad que se maneja es la función gaussiana y la distancia en el mapa es obtenida mediante la distancia de Manhattan. Además el mapa de neuronas se inicializa aleatoriamente con una distribución normal tomando como media y varianza, las del conjunto de datos de entrada. De esta forma se acelera la convergencia del proceso.

```
# Clase que modula un algoritmo SOM para clustering con restricciones.
# Contiene los metodos comunes y sirve de superclase para las clases
# de los algoritmos.
class SOMCC(object)

# Tiene los siguientes metodos:

# Crea el objeto SOM con una dimension shape que se pasa como
# una tupla y un valor de alfa inicial alpha_start flotante.
# Ademas da la posibilidad de definir una semilla.
def __init__(self, shape, alpha_start=0.6, seed=None)
```

```

# A partir de un vector de datos data de tamaño n y una matriz
# cuadrada de restricciones restr de tamaño n x n, se inicializan
# las neuronas del mapa de forma aleatoria distribuyendo sus pesos
# según una normal a partir de la media y desviación típica del
# vector data.
def initialize(self, data, restr)

# Entrena el mapa SOM con el conjunto de datos data y la matriz de
# restricciones matrixConstr, un número de épocas epochs. El
# entrenamiento se produce llamando internamente a cycle.
def fit(self, data, restr, matrixConstr)

# Calcula el número de restricciones que se incumplen al asignar la
# instancia con índice index en la lista de instancias asociadas a
# la neurona neuron pasando como parámetro la matriz de restricciones
# matrixConstr.
def incrementInfeasibility(self, neuron, index, matrixConstr)

# Crea los clusters asociados a las neuronas una vez terminado el
# entrenamiento de la red, obteniendo el centroide y la desviación
# cuadrática de cada cluster. Necesita recibir como parámetro el
# conjunto de datos data.
def build_map_clusters(self, data)

# Dados dos clusters cl1 y cl2 y la matriz de restricciones
# matrixConstr, obtiene el número de restricciones que se
# incumplirían si agrupamos los dos clusters.
def incrementInfeasibilityIC(self, matrixConstr, cl1, cl2)

# Tomando como parámetros el vector de datos, la matriz de
# restricciones matrixConstr y un entero k; se genera una
# partición de k clusters del conjunto, a partir de los
# clusters definidos por el mapa después de haber sido
# entrenado.
def generateKClusters(self, data, matrixConstr, k)

# Calcula el error medio de ajuste al mapa.
def som_error(self, data)

# Calcula la infactibilidad total de la solución al clustering,
# pasando como parámetro una lista de restricciones listConstr.
# En mode podemos definir si queremos 'cluster' o 'som'.
# En el primero se coge la infactibilidad de los clusters finales
# generados y el segundo de los clusters generados con el mapa SOM.
def calcularInfeasibilityTotal(self, listConstr, mode)

```

10. Implementación

```
# Calcula la distancia intra-cluster general de la particion solucion
# pasando como parametro el vector de datos data. En mode podemos
# definir si queremos ''cluster'' o ''som''. En el primero se calcula
# la distancia de los clusters finales generados y el segundo de los
# clusters generados con el mapa SOM.
def calcularDistICTotal(self, data, mode)

# Devuelve la asignacion de instancias a clusters.
def dataInCluster(self)
```

10.1.3. SCCSOM

Esta clase hereda de la clase general SOMCC para contar con todos los métodos comunes a un algoritmo SOM para clustering con restricciones. Implementa como métodos propios del algoritmo un criterio de elección de mejor neurona y el proceso de adaptación del SOM con una entrada.

```
# Clase que hereda de la clase SOMCC y define el algoritmo SCCSOM.
class SCCSOM(object)

# A partir de una instancia x, su posicion en el vector de entradas
# index y la matriz de restricciones matrixConstr restr se obtiene
# la mejor neurona en el mapa para la entrada x minimizando el
# incumplimiento de restricciones.
def softWinner(self, x, index, matrixConstr)

# Produce una iteracion en la adaptacion del SOM, para una entrada
# x con indice index. Llama a softWinner() para x, lo asocia a la
# lista de su neurona ganadora, ajusta los pesos del vecindario y
# finalmente actualiza los parametros alfa y sigma segun la epoca.
# Necesita como entrada una matriz de restricciones, matrixConstr
# y la epoca epoch.
def cycleSCCSOM(self, x, index, epoch, matrixConstr)
```

10.1.4. CS2S

Esta clase hereda de la clase general SOMCC como hacía SCCSOM. Implementa como métodos propios del algoritmo un criterio de elección de mejores neuronas en el entrenamiento y otro para elegir cluster final y el proceso de adaptación del SOM con dos entradas.

```
# Clase que hereda de la clase SOMCC y define el algoritmo CS2S.
class CS2S(object)
```

```

# Tomando como entrada dos instancias x1 y x2 y el valor de restriccion
# que tienen constr, obtiene las mejores neuronas para x1 y x2, con el
# criterio de eleccion de neuronas en el entrenamienti de CS2S.
def CS2SWinners(self, x1, x2, constr)

# Devuelve la neurona mas cercana a una instancia x.
def winner(self, x)

# Produce una iteracion en la adaptacion del SOM, para dos entradas
# x1 y x2 con indices index1 e index2. Llama a CS2SWinners, ajusta
# los pesos del vecindario y finalmente actualiza los parametros
# alfa y sigma segun la epoca.
# Necesita como entrada una matriz de restricciones, matrixConstr
# y la epoca epoch.
def cycleCS2S(self, x1, x2, index1, index2, epoch, matrixConstr)

```

10.1.5. CopSoft-SOM

Esta clase hereda de la clase general SOMCC como los dos métodos anteriores. Añade una inicialización a un agrupamiento aleatorio, un criterio de elección de mejor neurona, un proceso de satisfacción de restricciones y el proceso de adaptación del SOM con una entrada.

```

# Funcion que crea una asignacion de clusters aleatoria del conjunto
# de datos data con un numero de clusters k.
def randomClustering(data, k)

# Clase que hereda de la clase SOMCC y define el algoritmo CopSoft-SOM.
class CopSoft_SOM(object)

# A partir de una instancia x, su posicion en el vector de entradas
# index y la matriz de restricciones matrixConstr restr se obtiene
# la mejor neurona en el mapa para la entrada x y el vector S de
# instancias que se tienen que recolocar por entrar en conflicto
# con x.
def hardWinner(self, x, index, matrixConstr)

# Hace una asignacion minimizando restricciones de la instancia s con
# indice index. Necesita como entrada la matriz de restricciones.
def constraintViolation(self, s, indice, matrixConstr)

# Produce una iteracion en la adaptacion del SOM, para una entrada
# x con indice index. Llama a hardWinner() para x, lo asocia a la
# lista de su neurona ganadora, ajusta los pesos del vecindario,
# llama a constraintViolation y reasigna las instancias en conflicto
# y finalmente actualiza los parametros alfa y sigma segun la epoca.
# Necesita como entrada una matriz de restricciones, matrixConstr
# y la epoca epoch.
def cycleCopSoftSOM(self, x, index, epoch, matrixConstr)

```

10.1.6. Funciones y métodos comunes de ensemble para SCCSOM

Esta clase hereda de la clase general SCCSOM modificando su criterio de barajar el vector de entradas antes de cada época, y presentándole de entrada un vector ordenado según un criterio. Añade una función que elige las n mejores soluciones de la población, y como métodos una inicialización a partir de una población de soluciones, un nuevo método fit y un método para crear la matriz de coasociación.

```
# Funcion que elige las n mejores soluciones al agrupamiento dentro
# de una poblacion de soluciones , donde inf es el vector que almacena
# los valores de infactibilidad de las soluciones de pobl.
chooseBest(pobl, inf, n)

# Clase que modula un metodo basado en SCCSOM para ensemble. Contiene
# los metodos comunes y sirve de superclase para las clases de los
# algoritmos.
class SCCSOMEnsemble(object)

# Se sobrecarga el metodo initialize para que se pase como parametro
# el conjunto de soluciones al agrupamiento en vez del conjunto de
# datos. En el metodo se llama a buildCoasoaciationMatrix y define
# el orden del conjunto de instancias.
def initialize(self, pobl, restr)

# Se sobrecarga el metodo fit para entrenar el mapa sin barajar el
# vector de instancias en cada epoca. Lo demas se mantiene como en
# el metodo de la clase padre.
def fit(self, data, restr, matrixConstr)

# Metodo que crea la matriz de coasociacion para el ensemble.
def buildCoasoaciationMatrix(self)
```

10.1.7. SCCSOM-E

Esta clase hereda de SCCSOMEnsemble, ya que esta posee los métodos comunes para utilizar SOM en un método de “ensemble”. Además añade un método para calcular el vector con la certeza de cada instancia, y un método para ordenar la población con ese criterio.

```
# Clase que modula un metodo basado en SOM para clustering con
# restricciones con ensembles. Contiene los metodos comunes y
# sirve de superclase para las clases de los algoritmos.
class SCCSOM_E(object)

# Metodo que crea el vector que almacena la certeza de cada
# instancia.
def buildCertainly(self)
```



```
# Metodo que ordena el conjunto de instancias segun la certeza.
def sortPobl(self)
```

10.1.8. SCCSOM-EAC-SL

Esta clase hereda de SCCSOMEnsemble también, y su única modificación es que incluye un método para ordenar la población con el criterio de este algoritmo.

```
# Clase que modula un metodo basado en SOM para clustering con
# restricciones con ensembles. Contiene los metodos comunes y
# sirve de superclase para las clases de los algoritmos.
class SCCSOM_EAC_SL(object)

# Metodo que ordena el conjunto de instancias segun el criterio
# de SCCSOM-EAC-SL.
def sortPobl(self)
```


Parte IV.

Experimentación

11. Configuración experimental

Para nuestros experimentos compararemos los resultados obtenidos por los algoritmos propuestos sobre 31 conjuntos de datos y 3 conjuntos de restricciones (4 si se incluye el conjunto sin restricciones) para cada uno de ellos. Los conjuntos de datos se pueden encontrar en el repositorio [Keel-dataset repository](https://scizs.ugr.es/keel/category.php?cat=clas)¹ [TGM⁺17].

Nombre	Número de Instancias	Número de Atributos	Número de Clases
<i>Appendicitis</i>	106	7	2
<i>Balance</i>	625	4	3
<i>Banana</i>	1590	2	2
<i>Breast_cancer</i>	569	30	2
<i>Contraceptive</i>	1473	9	3
<i>Ecoli</i>	336	7	8
<i>Glass</i>	214	9	7
<i>Haberman</i>	306	3	2
<i>Hayes_roth</i>	160	4	3
<i>Heart</i>	270	13	2
<i>Ionosphere</i>	351	33	2
<i>Iris</i>	150	4	3
<i>Monk2</i>	432	6	6
<i>Newthyroid</i>	215	5	3
<i>Page_blocks</i>	1642	10	5
<i>Phoneme</i>	1622	5	2
<i>Pima</i>	768	8	2
<i>Saheart</i>	462	9	2
<i>Satimage</i>	1931	36	7
<i>Segment</i>	693	19	7
<i>Sonar</i>	208	60	2
<i>Soybean</i>	47	35	4
<i>Spambase</i>	1380	57	2
<i>Spectfheart</i>	267	44	2
<i>Tae</i>	151	5	3
<i>Thyroid</i>	2160	21	3
<i>Titanic</i>	661	3	2
<i>Vehicle</i>	846	18	4
<i>Wdbc</i>	569	30	2
<i>Wine</i>	178	13	3
<i>Zoo</i>	101	16	7

Tabla 11.1.: Resumen de los conjuntos de datos utilizados para los experimentos.

¹<https://scizs.ugr.es/keel/category.php?cat=clas>

11. Configuración experimental

Los conjuntos de datos de clasificación se utilizan comúnmente en la literatura para probar los algoritmos de clustering restringido; la razón detrás de esto es que nos permiten generar restricciones con respecto a las etiquetas verdaderas (véase la [Sección 11.1](#)). También facilitan una fácil evaluación de la calidad del algoritmo mediante medidas como el índice de Rand ajustado (véase la [Sección 11.2](#)).

11.1. Generación de restricciones

Conjunto de datos	CS10		CS15		CS20	
	ML	CL	ML	CL	ML	CL
<i>Appendicitis</i>	37	18	76	44	154	77
<i>Balance</i>	841	1112	1846	2525	3324	4426
<i>Banana_undersmpl</i>	6368	6193	14311	14130	25509	24894
<i>Breast_cancer</i>	846	750	1954	1701	3292	3149
<i>Contraceptive</i>	3893	6985	8620	15690	15372	27993
<i>Ecoli</i>	147	414	352	923	644	1634
<i>Glass</i>	58	173	138	390	233	670
<i>Haberman</i>	273	192	631	404	1173	718
<i>Hayes_roth</i>	47	73	86	190	177	319
<i>Heart</i>	173	178	436	384	747	684
<i>Ionosphere</i>	338	292	738	640	1357	1128
<i>Iris</i>	28	77	92	161	132	303
<i>Monk2</i>	484	462	1064	1016	1835	1906
<i>Newthyroid</i>	125	106	273	255	488	415
<i>Page_blocks_undersmpl</i>	11055	2475	24674	5707	43572	10384
<i>Phoneme_undersmpl</i>	7817	5386	17270	12376	30826	21824
<i>Pima</i>	1572	1354	3601	3069	6443	5338
<i>Saheart</i>	613	468	1281	1134	2368	1910
<i>Satimage_undersmpl</i>	3507	15214	7972	33933	14138	60553
<i>Segment_undersmpl</i>	377	2038	738	4618	1353	8238
<i>Sonar</i>	106	104	241	255	416	445
<i>Soybean</i>	1	9	11	17	5	40
<i>Spambase_undersmpl</i>	4969	4484	11075	10246	19904	18046
<i>Spectfheart</i>	230	121	563	257	952	479
<i>Tae</i>	31	89	88	165	164	301
<i>Thyroid_undersmpl</i>	19972	3248	45116	7210	79972	13124
<i>Titanic_undersmpl</i>	1249	962	2721	2229	4908	3870
<i>Vehicle</i>	916	2654	1993	6008	3576	10789
<i>Wdbc</i>	864	732	1975	1680	3448	2993
<i>Wine</i>	57	96	105	246	210	420
<i>Zoo</i>	13	42	27	93	53	157

Tabla 11.2.: Resumen de los conjuntos de restricciones utilizados para los experimentos.

Como dispongo de las etiquetas verdaderas asociadas a cada conjunto de datos, utilizaré el método propuesto en [WCRSo1], para generar conjuntos de restricciones artificiales. Este método consiste en seleccionar aleatoriamente dos instancias de un conjunto de datos, comparar sus etiquetas y, por último, establecer una restricción ML o CL en función de si las etiquetas son iguales o diferentes.

Generaré, para cada conjunto de datos, cuatro conjuntos diferentes de restricciones, CS0, CS10, CS15 y CS 20, que estarán asociados a cuatro pequeños porcentajes del tamaño del conjunto de datos: 0 %, 10 %, 15 % y 20 %. Siendo n_f la fracción del tamaño del conjunto de datos asociada a cada uno de estos porcentajes, la fórmula $\frac{n_f(n_f-1)}{2}$ nos indica cuántas restricciones artificiales se crearán para cada conjunto de restricciones.

La asignación aleatoria de restricciones tiene una ventaja potencial sobre la simple utilización de las restricciones contenidas en un grafo completo de n_f vértices: hay una menor probabilidad de sesgar el conjunto de restricciones para que tenga clases con poca representación. La Tabla 11.2 muestra el número de restricciones de cada tipo obtenidas para cada conjunto de datos. Obsérvese que cuanto mayor es el número de clases presentes en el conjunto de datos, menos restricciones ML se obtienen con el método propuesto en [WCRSo1]. Esto se debe a que la probabilidad de elegir al azar dos individuos de la misma clase disminuye a medida que aumenta el número de clases presentes en el conjunto de datos.

11.2. Método de evaluación

Dado que disponemos de las etiquetas verdaderas asociadas a cada uno de los conjuntos de datos, podemos utilizarlas para evaluar los resultados proporcionados por cada método. Utilizaremos el Índice de Rand ajustado (ARI) para medir la precisión de las predicciones resultantes de cada método que probamos. A continuación explicaré tanto el Índice de Rand ajustado como el Índice de Rand básico, que calcula el grado de acuerdo entre dos particiones C_1 y C_2 de un conjunto de datos X dado.

11.2.1. Índice de Rand

El índice o medida de Rand (llamado así por William M. Rand) es una medida de la similitud entre dos agrupaciones de datos que se define de la siguiente manera:

Definición 11.1. Dado un conjunto de n elementos $S = \{o_1, \dots, o_n\}$, y dos particiones de S para comparar, $X = \{X_1, \dots, X_r\}$, una partición de S en r subconjuntos, e $Y = \{Y_1, \dots, Y_s\}$, una partición de S en s subconjuntos. Se define lo siguiente:

- **a:** número de pares de elementos en S que están en el mismo subconjunto en X y en el mismo subconjunto en Y .
- **b:** número de pares de elementos en S que están en diferentes subconjuntos en X y en diferentes subconjuntos en Y .
- **c:** número de pares de elementos en S que están en el mismo subconjunto en X y en diferentes subconjuntos en Y .

11. Configuración experimental

- **d**: número de pares de elementos en S que están en diferentes subconjuntos en X y en el mismo subconjunto en Y .

Entonces, se define el índice de Rand, RI , como:

$$RI(X, Y) = \frac{a + b}{a + b + c + d}$$

Podemos considerar, $a + b$ como el número de concordancias y $c + d$ como el número de discrepancias entre X e Y . Como el denominador es el número total de pares, el índice de Rand representa la frecuencia de ocurrencia de concordancias sobre el número total de pares, o como la probabilidad de que X e Y concuerden en un par elegido aleatoriamente. Desde un punto de vista matemático, el índice de Rand está relacionado con la precisión.

Similarmente, se puede ver el RI como una medida del porcentaje de acierto del algoritmo con la siguiente fórmula:

$$RI(X, Y) = \frac{TP + TN}{TP + FP + FN + TN}$$

donde TP es el número de verdaderos positivos, TN el número de verdaderos negativos, FP es el número de falsos positivos y FN es el número de falsos negativos.

11.2.2. Índice de Rand ajustado

El índice de Rand ajustado, ARI , es la versión corregida por el azar del índice de Rand. Dicha corrección por azar establece una línea de base utilizando la similitud esperada de todas las comparaciones por pares entre agrupaciones especificadas por un modelo aleatorio.

Aunque el índice de Rand sólo puede dar un valor entre 0 y 1, el índice de Rand ajustado puede dar valores entre -1 y 1. Produce un valor negativo si el índice es menor que el índice esperado.

Dado un conjunto S de n elementos, y dos particiones de S , $X = \{X_1, X_2, \dots, X_r\}$ e $Y = \{Y_1, Y_2, \dots, Y_s\}$, el solapamiento entre X e Y puede resumirse en una tabla de contingencia $[n_{ij}]$ donde cada entrada n_{ij} denota el número de elementos en común entre X_i e Y_j :

$$n_{ij} = |X_i \cap Y_j|$$

$X \setminus Y$	Y_1	Y_2	\dots	Y_s	$\sum_{j=1}^s n_{ij}$
X_1	n_{11}	n_{12}	\dots	n_{1s}	$n_{1.}$
X_2	n_{21}	n_{22}	\dots	n_{2s}	$n_{2.}$
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
X_r	n_{r1}	n_{r2}	\dots	n_{rs}	$n_{r.}$
$\sum_{i=1}^r n_{ij}$	$n_{.1}$	$n_{.2}$	\dots	$n_{.s}$	n

Definición 11.2. Se define el Índice de Rand Ajustado, ARI , como:

$$ARI(X, Y) = \frac{RI(X, Y) - \text{Valor Esperado}}{\text{Total de pares} - \text{Valor Esperado}}$$

de donde se obtiene que

$$ARI(X, Y) = \frac{\sum_{ij} \binom{n_{ij}}{2} - \left[\sum_i \binom{n_{i.}}{2} \sum_j \binom{n_{.j}}{2} \right] / \binom{n}{2}}{\frac{1}{2} \left[\sum_i \binom{n_{i.}}{2} + \sum_j \binom{n_{.j}}{2} \right] - \left[\sum_i \binom{n_{i.}}{2} \sum_j \binom{n_{.j}}{2} \right] / \binom{n}{2}}$$

donde $n_{ij}, n_{i.}, n_{.j}$ son valores de la tabla de contingencia.

Nuestro objetivo es cuantificar la calidad de las soluciones obtenidas como resultado de los métodos presentados en este trabajo. Para llevar a cabo esta tarea, nos limitamos a establecer una de las dos particiones dadas para calcular el ARI como las etiquetas de verdad absoluta.

11.3. Elección de los valores de los hiperparámetros

En la [Tabla 11.3](#) se muestra un resumen del valor dado a los hiperparámetros de los algoritmos SCCSOM, CS2S y CopSoft-SOM; y en la [Tabla 11.4](#) también se muestra otro resumen de valores para los hiperparámetros, pero en este caso para los algoritmos de “ensemble”, SCCSOM-E y SCCSOM-EAC-SL. Los métodos de ensemble reciben como soluciones de entrada 20 soluciones obtenidas con el algoritmo SCCSOM cada una inicializada con una semilla distinta, concretamente los valores de semillas serán del 1 al 20. El conjunto de soluciones de entrada es el mismo para los dos métodos de “ensemble”.

Parámetro	Significado	Valor
max_epocas	Número de épocas que se entrenará el mapa.	30
alpha_start	Valor alpha inicial.	0.6
shape	Dimensión del mapa auto-organizativo.	(2,2,2) y (3,3) (2 experimentos)
k	Número de clusters	Dependiendo del conjunto de datos, Tabla 11.1 .

Tabla 11.3.: Configuración de los parámetros para SCCSOM, CS2S y CopSoft-SOM.

Parámetro	Significado	Valor
max_epocas	Número de épocas que se entrenará el mapa.	30
alpha_start	Valor alpha inicial.	0.6
shape	Dimensión del mapa autoorganizado.	(2,2,2) y (3,3) (2 experimentos)
k	Número de clusters	Dependiendo del conjunto de datos, Tabla 11.1 .
N	Tamaño de la población inicial de soluciones.	20
M	Tamaño final de la población de mejores soluciones.	15

Tabla 11.4.: Configuración de los parámetros para SCCSOM-E y SCCSOM-EAL-SL.

12. Resultados experimentales.

En esta sección presentamos las Tablas, que muestran los resultados obtenidos por los 5 algoritmos, los 3 básicos de SOM y los dos de “ensemble”. Como estos algoritmos utilizan procedimientos no deterministas para alcanzar una solución, los resultados pueden variar de una ejecución a otra. Para reducir el efecto que esto puede tener en los resultados, aplicaremos cada algoritmo 10 veces, con 10 semillas distintas, a cada conjunto de datos, cada conjunto de restricciones y cada topología del SOM, de modo que las medidas mostradas en las tablas mencionadas anteriormente respondan a la media de las 10 ejecuciones. Por tanto, con 31 conjuntos de datos, 4 conjuntos de restricciones (contamos el conjunto con el 0.0 % de restricciones) para cada uno de ellos, 5 métodos diferentes, 2 topologías de mapa y 30 ejecuciones, hemos realizado un total de 74400 experimentos para nuestro estudio. Hay que señalar que en estas tablas sólo aparecen los resultados medios, aunque si se desea en el [Apéndice A](#) se puede encontrar el desglose de estos resultados medios.

En la [Tabla 12.1](#) nos encontramos con los resultados producidos por el algoritmo SCC-SOM. Observando los resultados se puede concluir que este algoritmo escala muy bien conforme aumenta el número de restricciones, pasando de un valor medio de 0.1708 para el SOM 2D con el 0 % de restricciones a un 0.8926 con el 20 % de restricciones, y de 0.1903 con el 0 % a 0.8942 con el 20 % para el SOM3D. Además se aprecia como la desviación típica de los datos decrementa conforme aumentan las restricciones observando los datos para el 10 %, 15 % y 20 %. Salvo los conjuntos de datos Tae y Zoo, todos los demás mejoran con el aumento de restricciones y con este criterio de reducir la infactibilidad. Además se aprecia como aunque la topología 3D suele ganar, los resultados para ambas topologías son muy parecidos, asimilándose mucho su repuesta para varios conjunto de datos.

En la [Tabla 12.2](#) nos encontramos con los resultados producidos por el algoritmo TCS2S. Lo primero que se aprecia al observar los datos es que el algoritmo no escala conforme aumenta el número de restricciones, esto viene a consecuencia de que este algoritmo no almacena las instancias de datos en los clusters asociados a las neuronas, sino que toda su estrategia se basa en acercar una neurona al punto medio del espacio entre dos instancias que tengan una restricción ML y alejarla del punto medio de donde existan dos con una CL. Esta estrategia es muy pobre si pensamos en la estrategia de los algoritmos SCCSOM y CopSoft-SOM. Por tanto, la máxima influencia que consigue de las restricciones es aumentar entre un 0.5 y un 0.10 su ARI con respecto al problema de clustering sin restricciones. Se tiene que con un 10 % la estrategia de acercar y alejar ha dado su máximo efecto, y que aunque aumenten las restricciones no consigue mejorar el agrupamiento. En lo demás, en lo respectivo a las dos topologías se vuelven a presentar resultados muy parecidos y en lo relativo a los conjuntos de datos, todos presentan una respuesta parecida a la que presentarían con un agrupamiento sin restricciones.

En la [Tabla 12.3](#) nos encontramos con los resultados producidos por el algoritmo CopSoft-SOM, este algoritmo como le pasaba al primero también escala conforme aumentan las

12. Resultados experimentales.

restricciones, pero presenta resultados peores en la mayoría de los casos. Esto tiene sentido si se tiene en cuenta que este método prioriza intentar una asignación incumpliendo 0 restricciones. Cuando se va a provocar una violación reasigna todas las instancias con las que tiene un conflicto, y asigna esa instancia problemática al mejor cluster, esa asignación de la problemática a su mejor cluster sin tener en cuenta las restricciones es lo que provoca que este método sea peor que el primero. Se observa que con el 20% de restricciones es capaz de aumentar el ARI en 0.3 aproximadamente. Con respecto a las topologías, vuelven a presentar resultados parecidos, aunque sigue venciendo el SOM 3D.

En la [Tabla 12.4](#) y [Tabla 12.5](#) nos encontramos con los resultados producidos por los algoritmos de “ensemble”, SCCSOM-E y SCCSOM-EAC-SL. Estos dos métodos se tienen que comparar entre ellos, y no con los otros tres anteriores, ya que la naturaleza del problema es distinto. Observando los resultados se observa que con ambos se alcanzan buenas soluciones para el problema con el 20% de restricciones, esto tiene sentido porque están usando como entrada soluciones que sí alcanzan esos valores por separado, por lo que la solución obtenida a partir de ellas como entrada debe de alcanzar también estas buenas soluciones. La mayor diferencia entre los métodos se produce en los conjuntos con 0%, 10% y 15% de restricciones, mientras que la ordenación obtenida en el SOM-EAL-SL con pocas restricciones es capaz de presentar buenos resultados, incluso sin ninguna restricción en el “ensemble”, la ordenación obtenida en SCCSOM-E tiene un resultado más parecido al método SCCSOM, ya que se comporta igual para cada conjunto de restricciones. Por tanto se concluye que SCCSOM-EAL-SL es capaz de sacar mejor provecho a las soluciones de entrada que SCCSOM-E.

Nombre	0 % restricciones		10 % restricciones		15 % restricciones		20 % restricciones	
	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D
Appendicitis	0.5164 ± 0.0247	0.4975 ± 0.0343	0.2604 ± 0.1938	0.2674 ± 0.2882	0.3698 ± 0.2437	0.4776 ± 0.1948	0.6554 ± 0.2861	0.8308 ± 0.135
Balance	0.0607 ± 0.0177	0.0578 ± 0.0312	0.7973 ± 0.1238	0.8282 ± 0.0482	0.9874 ± 0.0242	0.9951 ± 0.0028	0.9771 ± 0.0362	0.9595 ± 0.0532
Banana	0.003 ± 0.0036	0.0437 ± 0.0306	0.8983 ± 0.2993	0.899 ± 0.2996	1.0 ± 0.0000	1.0 ± 0.0000	1.0 ± 0.0000	1.0 ± 0.0000
Breast_cancer	0.2824 ± 0.0212	0.3576 ± 0.0122	0.9581 ± 0.0127	0.9588 ± 0.0121	0.9986 ± 0.0028	0.9944 ± 0.0028	1.0 ± 0.0000	1.0 ± 0.0000
Contraceptive	0.0118 ± 0.001	0.022 ± 0.0011	0.9924 ± 0.0045	0.991 ± 0.0015	1.0 ± 0.0000	1.0 ± 0.0000	1.0 ± 0.0000	1.0 ± 0.0000
Ecoli	0.2111 ± 0.0083	0.2989 ± 0.0142	0.2101 ± 0.0851	0.232 ± 0.053	0.79 ± 0.0343	0.7207 ± 0.155	0.9134 ± 0.0355	0.9405 ± 0.0157
Glass	0.213 ± 0.0202	0.2307 ± 0.0171	0.1476 ± 0.0397	0.1485 ± 0.0158	0.2831 ± 0.1516	0.3482 ± 0.1317	0.7827 ± 0.0622	0.7478 ± 0.0923
Haberman	-0.0136 ± 0.0109	-0.0247 ± 0.0106	0.166 ± 0.1323	0.3044 ± 0.0944	0.9182 ± 0.2453	0.9825 ± 0.0525	0.8817 ± 0.2268	0.8678 ± 0.295
Hayes_roth	0.0802 ± 0.0223	0.0224 ± 0.0285	0.0438 ± 0.0416	0.0466 ± 0.0481	0.219 ± 0.1201	0.186 ± 0.1017	0.6017 ± 0.1228	0.6018 ± 0.1112
Heart	0.0033 ± 0.0034	0.0085 ± 0.0036	0.0806 ± 0.082	0.2005 ± 0.1338	0.6773 ± 0.2835	0.8739 ± 0.2902	0.9219 ± 0.2343	0.9141 ± 0.2576
Ionosphere	0.0106 ± 0.0894	0.0019 ± 0.0192	0.356 ± 0.1823	0.5758 ± 0.12	0.8829 ± 0.2936	0.9888 ± 0.0336	1.0 ± 0.0000	1.0 ± 0.0000
Iris	0.6319 ± 0.0697	0.7542 ± 0.0716	0.6493 ± 0.1075	0.6807 ± 0.1283	0.7834 ± 0.1402	0.8374 ± 0.0994	0.8977 ± 0.0914	0.9045 ± 0.0633
Monk2	0.0965 ± 0.0423	0.1144 ± 0.0368	0.5273 ± 0.1082	0.5035 ± 0.0953	0.9805 ± 0.0027	0.9817 ± 0.0017	0.9533 ± 0.0823	0.9818 ± 0.0005
Newthyroid	0.5108 ± 0.1137	0.4953 ± 0.0325	0.2706 ± 0.1131	0.4055 ± 0.1511	0.8551 ± 0.0413	0.831 ± 0.0544	0.9215 ± 0.116	0.9605 ± 0.0253
Page_blocks	-0.009 ± 0.0042	-0.0253 ± 0.0022	0.9851 ± 0.0006	0.9856 ± 0.0005	0.9907 ± 0.0033	0.9904 ± 0.0031	0.9968 ± 0.0023	0.9968 ± 0.0017
Phoneme	0.2188 ± 0.0104	0.2613 ± 0.0097	0.8982 ± 0.2977	0.8977 ± 0.2975	1.0 ± 0.0000	1.0 ± 0.0000	1.0 ± 0.0000	1.0 ± 0.0000
Pima	0.0473 ± 0.0045	-0.0009 ± 0.0004	0.9892 ± 0.0274	0.9974 ± 0.0026	0.9958 ± 0.0021	0.9968 ± 0.0026	1.0 ± 0.0000	1.0 ± 0.0000
Saheart	-0.0035 ± 0.0165	-0.0079 ± 0.0081	0.7163 ± 0.2285	0.8227 ± 0.045	0.5071 ± 0.4222	0.9835 ± 0.0099	0.944 ± 0.1679	1.0 ± 0.0000
Satimage	0.3654 ± 0.0036	0.4634 ± 0.013	0.9747 ± 0.0079	0.9746 ± 0.0134	0.9989 ± 0.0008	0.9985 ± 0.0056	0.9985 ± 0.0008	0.9992 ± 0.0005
Segment	0.2303 ± 0.0319	0.2606 ± 0.0102	0.1436 ± 0.0558	0.1808 ± 0.0565	0.6213 ± 0.1895	0.7095 ± 0.1906	0.9452 ± 0.0503	0.9897 ± 0.0146
Sonar	0.0002 ± 0.0068	-0.0045 ± 0.0003	0.071 ± 0.083	0.0734 ± 0.0983	0.492 ± 0.2155	0.478 ± 0.2416	0.7801 ± 0.3927	0.6542 ± 0.4349
Soybean	0.5862 ± 0.0146	0.5484 ± 0.0442	0.6002 ± 0.138	0.5805 ± 0.0946	0.6452 ± 0.1736	0.6629 ± 0.1398	0.4934 ± 0.0658	0.46 ± 0.0764
Spambase	0.0606 ± 0.0035	0.0732 ± 0.0035	0.8952 ± 0.2978	0.9945 ± 0.0009	1.0 ± 0.0000	1.0 ± 0.0000	1.0 ± 0.0000	1.0 ± 0.0000
Spectfheart	-0.0997 ± 0.0022	-0.105 ± 0.001	0.4049 ± 0.1527	0.31 ± 0.1565	0.9558 ± 0.082	0.8644 ± 0.266	0.9762 ± 0.0397	0.9073 ± 0.2615
Tae	0.0057 ± 0.0041	0.0203 ± 0.0172	0.0286 ± 0.0195	0.014 ± 0.023	0.0771 ± 0.0415	0.0614 ± 0.0409	0.5868 ± 0.121	0.5706 ± 0.1245
Thyroid	-0.0251 ± 0.0118	-0.0077 ± 0.001	0.9901 ± 0.0009	0.9896 ± 0.001	0.995 ± 0.0023	0.9969 ± 0.0013	0.9431 ± 0.1594	0.9992 ± 0.0004
Titanic	0.0512 ± 0.0814	0.1484 ± 0.0484	0.9795 ± 0.041	0.9926 ± 0.0024	0.9914 ± 0.0049	0.9914 ± 0.0073	1.0 ± 0.0000	1.0 ± 0.0000
Vehicle	0.0894 ± 0.0053	0.0695 ± 0.0036	0.4089 ± 0.0927	0.4424 ± 0.1433	0.9513 ± 0.1022	0.9857 ± 0.0028	1.0000 ± 0.0000	1.0000 ± 0.0000
Wdbc	0.2824 ± 0.0212	0.3576 ± 0.0122	0.9782 ± 0.011	0.9831 ± 0.0109	1.0000 ± 0.0000	0.9958 ± 0.0035	1.0000 ± 0.0000	1.0000 ± 0.0000
Wine	0.2147 ± 0.0112	0.332 ± 0.0134	0.3277 ± 0.0987	0.3487 ± 0.0988	0.5114 ± 0.0998	0.4538 ± 0.0951	0.8571 ± 0.1165	0.9013 ± 0.0701
Zoo	0.6616 ± 0.0267	0.6368 ± 0.018	0.4813 ± 0.0849	0.439 ± 0.0791	0.4699 ± 0.1003	0.5234 ± 0.0936	0.6423 ± 0.1099	0.5335 ± 0.1389
Media	0.1708 ± 0.2111	0.1903 ± 0.225	0.5558 ± 0.3506	0.5829 ± 0.3436	0.7725 ± 0.2724	0.8044 ± 0.2672	0.8926 ± 0.1446	0.8942 ± 0.1583

Tabla 12.1.: Resultados medios de las ejecuciones del SCCSOM.

Nombre	0 % restricciones		10 % restricciones		15 % restricciones		20 % restricciones	
	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D
Appendicitis	0.5306 ± 0.0153	0.5041 ± 0.0236	0.456 ± 0.0304	0.4142 ± 0.0428	0.4405 ± 0.037	0.4513 ± 0.0184	0.4147 ± 0.0448	0.4839 ± 0.026
Balance	0.0736 ± 0.0149	0.0571 ± 0.041	0.1443 ± 0.0508	0.2914 ± 0.0616	0.2711 ± 0.1123	0.3253 ± 0.1061	0.1122 ± 0.0298	0.1539 ± 0.0435
Banana	0.0009 ± 0.0013	0.0017 ± 0.0043	0.1194 ± 0.0149	0.1815 ± 0.072	0.0885 ± 0.0367	0.1751 ± 0.0342	0.0947 ± 0.0245	0.175 ± 0.0184
Breast_cancer	0.3038 ± 0.0246	0.3614 ± 0.0356	0.6369 ± 0.0146	0.6377 ± 0.0255	0.6369 ± 0.022	0.607 ± 0.011	0.6399 ± 0.019	0.6485 ± 0.0232
Contraceptive	0.0107 ± 0.0017	0.0232 ± 0.0026	0.0185 ± 0.0099	0.0251 ± 0.0008	0.0195 ± 0.0079	0.0244 ± 0.0044	0.0245 ± 0.0035	0.0266 ± 0.001
Ecoli	0.2275 ± 0.0262	0.2784 ± 0.0407	0.4107 ± 0.0198	0.3189 ± 0.0255	0.4558 ± 0.0142	0.4066 ± 0.0169	0.4103 ± 0.0546	0.3522 ± 0.0445
Glass	0.2255 ± 0.0066	0.2268 ± 0.0148	0.2345 ± 0.0078	0.2379 ± 0.0133	0.245 ± 0.0123	0.2526 ± 0.0186	0.2438 ± 0.0206	0.2675 ± 0.0242
Haberman	-0.0212 ± 0.0161	-0.0225 ± 0.0097	0.2027 ± 0.0107	0.2148 ± 0.0131	0.1487 ± 0.0299	0.187 ± 0.0533	0.1889 ± 0.0221	0.1702 ± 0.0441
Hayes_roth	0.0776 ± 0.0152	0.0364 ± 0.0348	0.0215 ± 0.0144	0.0345 ± 0.0172	0.0669 ± 0.0396	0.0818 ± 0.0247	0.089 ± 0.0637	0.1047 ± 0.0347
Heart	0.0022 ± 0.0013	0.0071 ± 0.0028	0.0448 ± 0.0261	0.041 ± 0.0277	0.0978 ± 0.0257	0.0935 ± 0.015	0.0837 ± 0.0178	0.0813 ± 0.0077
Ionosphere	-0.0153 ± 0.0126	0.0245 ± 0.0246	0.2746 ± 0.1119	0.2476 ± 0.0742	0.2933 ± 0.0812	0.289 ± 0.0871	0.3571 ± 0.0402	0.241 ± 0.0303
Iris	0.6166 ± 0.067	0.695 ± 0.032	0.6691 ± 0.0208	0.7116 ± 0.0598	0.7424 ± 0.0946	0.7576 ± 0.1071	0.7082 ± 0.0067	0.7555 ± 0.0765
Monk2	0.0687 ± 0.0498	0.1242 ± 0.0349	0.262 ± 0.0614	0.2794 ± 0.0835	0.2939 ± 0.0886	0.2221 ± 0.0594	0.3122 ± 0.0371	0.2681 ± 0.0664
Newthyroid	0.4271 ± 0.0834	0.3569 ± 0.1483	0.4329 ± 0.0915	0.3621 ± 0.0197	0.5516 ± 0.0709	0.5183 ± 0.0307	0.4817 ± 0.0984	0.4004 ± 0.0805
Page_blocks	-0.0202 ± 0.0066	-0.0208 ± 0.0085	0.0626 ± 0.0128	0.0459 ± 0.0458	0.0389 ± 0.0023	0.0486 ± 0.0472	0.0902 ± 0.0341	0.0221 ± 0.0168
Phoneme	0.0744 ± 0.1434	0.2774 ± 0.0052	0.2878 ± 0.0116	0.2568 ± 0.0552	0.2854 ± 0.0027	0.2379 ± 0.0514	0.2599 ± 0.0432	0.2502 ± 0.0443
Pima	0.0461 ± 0.0029	0.0328 ± 0.0334	0.0753 ± 0.0426	0.0701 ± 0.0276	0.1041 ± 0.0401	0.0897 ± 0.0178	0.1227 ± 0.048	0.0725 ± 0.0241
Saheart	-0.0167 ± 0.015	-0.0039 ± 0.0077	0.068 ± 0.0047	0.0752 ± 0.0086	0.0865 ± 0.0144	0.0787 ± 0.0088	0.069 ± 0.0154	0.0777 ± 0.0125
Satimage	0.3753 ± 0.0161	0.4815 ± 0.035	0.3885 ± 0.0233	0.4908 ± 0.0525	0.3611 ± 0.0284	0.4464 ± 0.011	0.3744 ± 0.0286	0.4717 ± 0.0365
Segment	0.2562 ± 0.0129	0.2706 ± 0.0141	0.2683 ± 0.0232	0.2747 ± 0.0177	0.2752 ± 0.0138	0.2868 ± 0.0517	0.2782 ± 0.0278	0.2806 ± 0.0288
Sonar	-0.0044 ± 0.0003	-0.003 ± 0.0024	0.0139 ± 0.0164	0.0081 ± 0.0114	0.008 ± 0.0066	-0.0003 ± 0.0065	0.0171 ± 0.0127	0.0284 ± 0.0188
Soybean	0.5842 ± 0.0256	0.5414 ± 0.0343	0.5846 ± 0.019	0.5472 ± 0.0124	0.6236 ± 0.0787	0.5215 ± 0.0767	0.5085 ± 0.0598	0.5538 ± 0.0197
Spambase	0.0473 ± 0.0038	0.0734 ± 0.0008	0.1642 ± 0.0144	0.1493 ± 0.005	0.153 ± 0.0027	0.1545 ± 0.0059	0.158 ± 0.0101	0.1526 ± 0.0051
Spectfheart	-0.0974 ± 0.0045	-0.1056 ± 0.0009	0.294 ± 0.037	0.2542 ± 0.0377	0.2149 ± 0.0174	0.2921 ± 0.0093	0.2568 ± 0.0174	0.2787 ± 0.0341
Tae	0.008 ± 0.0059	0.0008 ± 0.0099	0.019 ± 0.0101	0.0484 ± 0.0178	0.0314 ± 0.0242	0.0283 ± 0.0126	0.0252 ± 0.0117	0.0427 ± 0.0206
Thyroid	-0.0138 ± 0.0124	-0.0118 ± 0.0056	-0.0247 ± 0.002	-0.0177 ± 0.0116	-0.0104 ± 0.0257	-0.0232 ± 0.0015	-0.0185 ± 0.0116	-0.0245 ± 0.002
Titanic	0.0781 ± 0.0889	0.1384 ± 0.0773	0.2209 ± 0.0052	0.225 ± 0.0095	0.161 ± 0.0095	0.2271 ± 0.007	0.1595 ± 0.0941	0.225 ± 0.0095
Vehicle	0.1033 ± 0.0078	0.0773 ± 0.0099	0.1379 ± 0.011	0.0919 ± 0.0057	0.121 ± 0.0275	0.1024 ± 0.0296	0.1292 ± 0.0121	0.0848 ± 0.0068
Wdbc	0.3038 ± 0.0246	0.3614 ± 0.0356	0.6382 ± 0.0177	0.6077 ± 0.0163	0.619 ± 0.0118	0.6153 ± 0.0264	0.6376 ± 0.0189	0.6122 ± 0.0145
Wine	0.2438 ± 0.0511	0.3453 ± 0.0123	0.3934 ± 0.0245	0.3885 ± 0.0132	0.3989 ± 0.0053	0.4213 ± 0.0258	0.4299 ± 0.0102	0.4017 ± 0.0248
Zoo	0.674 ± 0.0172	0.6338 ± 0.0172	0.6642 ± 0.0216	0.6484 ± 0.0146	0.6899 ± 0.0095	0.7551 ± 0.0153	0.8103 ± 0.0018	0.7157 ± 0.0837
Media	0.1668 ± 0.2113	0.1859 ± 0.2158	0.264 ± 0.2111	0.2633 ± 0.206	0.2746 ± 0.22	0.2798 ± 0.2172	0.2732 ± 0.2186	0.2701 ± 0.2158

Tabla 12.2.: Resultados medios de las ejecuciones del CS2S.

Nombre	0 % restricciones		10 % restricciones		15 % restricciones		20 % restricciones	
	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D
Appendicitis	0.5164 ± 0.0247	0.4975 ± 0.0343	0.2331 ± 0.0204	0.0809 ± 0.1275	0.5425 ± 0.0573	0.5626 ± 0.0981	0.5524 ± 0.1083	0.459 ± 0.1134
Balance	0.0607 ± 0.0177	0.0578 ± 0.0312	0.4668 ± 0.0486	0.4866 ± 0.0439	0.5149 ± 0.0378	0.5204 ± 0.0478	0.5486 ± 0.0879	0.5946 ± 0.0357
Banana	0.003 ± 0.0036	0.0437 ± 0.0306	0.2805 ± 0.0332	0.4254 ± 0.1078	0.5493 ± 0.0538	0.5567 ± 0.1134	0.6192 ± 0.1082	0.6486 ± 0.1009
Breast_cancer	0.2824 ± 0.0212	0.3576 ± 0.0122	0.7298 ± 0.0315	0.7669 ± 0.0365	0.755 ± 0.0277	0.8026 ± 0.0246	0.7974 ± 0.0299	0.822 ± 0.0264
Contraceptive	0.0118 ± 0.001	0.022 ± 0.0011	0.1226 ± 0.0224	0.1671 ± 0.016	0.4906 ± 0.0318	0.5147 ± 0.0453	0.6234 ± 0.0396	0.6892 ± 0.0271
Ecoli	0.2111 ± 0.0083	0.2989 ± 0.0142	0.3373 ± 0.0328	0.2952 ± 0.031	0.5178 ± 0.0542	0.4706 ± 0.0556	0.4985 ± 0.0476	0.5612 ± 0.0615
Glass	0.213 ± 0.0202	0.2307 ± 0.0171	0.138 ± 0.036	0.1327 ± 0.032	0.2135 ± 0.0412	0.2532 ± 0.0195	0.2785 ± 0.0398	0.2943 ± 0.0582
Haberman	-0.0136 ± 0.0109	-0.0247 ± 0.0106	0.3254 ± 0.0376	0.3297 ± 0.0614	0.3851 ± 0.0567	0.3942 ± 0.0523	0.3576 ± 0.0597	0.405 ± 0.055
Hayes_roth	0.0802 ± 0.0223	0.0224 ± 0.0285	0.0359 ± 0.036	0.0267 ± 0.0275	0.1354 ± 0.0742	0.1346 ± 0.0438	0.1419 ± 0.0447	0.1277 ± 0.0784
Heart	0.0033 ± 0.0034	0.0085 ± 0.0036	0.1509 ± 0.0512	0.0976 ± 0.0745	0.136 ± 0.0728	0.1623 ± 0.0393	0.1981 ± 0.044	0.2311 ± 0.019
Ionosphere	0.0106 ± 0.0894	0.0019 ± 0.0192	0.3568 ± 0.1386	0.3059 ± 0.0354	0.5524 ± 0.0619	0.471 ± 0.0726	0.6364 ± 0.0383	0.5948 ± 0.06
Iris	0.6319 ± 0.0697	0.7542 ± 0.0716	0.4354 ± 0.0716	0.4322 ± 0.0756	0.8573 ± 0.0759	0.8511 ± 0.0762	0.9136 ± 0.0912	0.9512 ± 0.0367
Monk2	0.0965 ± 0.0423	0.1144 ± 0.0368	0.2764 ± 0.0964	0.3202 ± 0.0614	0.2498 ± 0.0754	0.3051 ± 0.0552	0.3521 ± 0.0975	0.3728 ± 0.0511
Newthyroid	0.5108 ± 0.1137	0.4953 ± 0.0325	0.4478 ± 0.0787	0.3481 ± 0.0718	0.6169 ± 0.0648	0.4817 ± 0.0596	0.6864 ± 0.0548	0.5359 ± 0.038
Page_blocks	-0.009 ± 0.0042	-0.0253 ± 0.0022	0.2165 ± 0.0164	0.2174 ± 0.042	0.3233 ± 0.0345	0.2994 ± 0.0381	0.3788 ± 0.0573	0.3215 ± 0.0663
Phoneme	0.2188 ± 0.0104	0.2613 ± 0.0097	0.5217 ± 0.0221	0.5813 ± 0.0411	0.7122 ± 0.0284	0.7818 ± 0.0282	0.7719 ± 0.06	0.8307 ± 0.0484
Pima	0.0473 ± 0.0045	-0.0009 ± 0.0004	0.2715 ± 0.0415	0.262 ± 0.0499	0.4481 ± 0.0415	0.4705 ± 0.0632	0.5694 ± 0.0316	0.5986 ± 0.0799
Saheart	-0.0035 ± 0.0165	-0.0079 ± 0.0081	0.1341 ± 0.0251	0.1627 ± 0.0346	0.1983 ± 0.0192	0.2174 ± 0.0349	0.305 ± 0.0334	0.3606 ± 0.0382
Satimage	0.3654 ± 0.0036	0.4634 ± 0.013	0.744 ± 0.0082	0.8213 ± 0.0205	0.8204 ± 0.0143	0.8986 ± 0.0117	0.8444 ± 0.0422	0.9205 ± 0.026
Segment	0.2303 ± 0.0319	0.2606 ± 0.0102	0.2875 ± 0.0263	0.2911 ± 0.0271	0.3734 ± 0.0391	0.3958 ± 0.0625	0.639 ± 0.0508	0.6725 ± 0.0893
Sonar	0.0002 ± 0.0068	-0.0045 ± 0.0003	0.0233 ± 0.0218	0.0255 ± 0.0252	0.0381 ± 0.0284	0.0283 ± 0.0249	0.0646 ± 0.0393	0.1055 ± 0.0384
Soybean	0.5862 ± 0.0146	0.5484 ± 0.0442	0.5588 ± 0.0866	0.5923 ± 0.0881	0.7601 ± 0.2072	0.7496 ± 0.1661	0.4928 ± 0.1424	0.4514 ± 0.0567
Spambase	0.0606 ± 0.0035	0.0732 ± 0.0035	0.3822 ± 0.0289	0.4478 ± 0.035	0.5949 ± 0.0269	0.5978 ± 0.0327	0.5642 ± 0.1409	0.663 ± 0.1058
Spectfheart	-0.0997 ± 0.0022	-0.105 ± 0.001	0.1476 ± 0.0498	0.1131 ± 0.0511	0.2418 ± 0.1063	0.2361 ± 0.1027	0.258 ± 0.123	0.2677 ± 0.1195
Tae	0.0057 ± 0.0041	0.0203 ± 0.0172	0.017 ± 0.0162	0.0332 ± 0.0368	0.0245 ± 0.0289	0.0176 ± 0.0097	0.0298 ± 0.0233	0.0307 ± 0.0257
Thyroid	-0.0251 ± 0.0118	-0.0077 ± 0.001	0.299 ± 0.2172	0.774 ± 0.0351	0.4168 ± 0.2183	0.9268 ± 0.0491	0.3631 ± 0.2177	0.9128 ± 0.116
Titanic	0.0512 ± 0.0814	0.1484 ± 0.0484	0.4293 ± 0.0343	0.4654 ± 0.0284	0.555 ± 0.047	0.563 ± 0.0408	0.677 ± 0.0387	0.7197 ± 0.0513
Vehicle	0.0894 ± 0.0053	0.0695 ± 0.0036	0.1215 ± 0.0245	0.1127 ± 0.0303	0.3524 ± 0.0339	0.3733 ± 0.0665	0.5551 ± 0.0516	0.6141 ± 0.0264
Wdbc	0.2824 ± 0.0212	0.3576 ± 0.0122	0.7698 ± 0.0261	0.7936 ± 0.0152	0.7507 ± 0.0288	0.7682 ± 0.0307	0.8272 ± 0.0245	0.8304 ± 0.0247
Wine	0.2147 ± 0.0112	0.332 ± 0.0134	0.311 ± 0.0718	0.2637 ± 0.0916	0.4717 ± 0.0636	0.488 ± 0.0545	0.6731 ± 0.0597	0.6814 ± 0.0512
Zoo	0.6575 ± 0.035	0.6476 ± 0.0151	0.3783 ± 0.0863	0.355 ± 0.0394	0.4694 ± 0.1216	0.4698 ± 0.0748	0.6145 ± 0.0931	0.6477 ± 0.1238
Media	0.1707 ± 0.2108	0.1907 ± 0.2257	0.321 ± 0.1976	0.3396 ± 0.2309	0.4538 ± 0.2228	0.4762 ± 0.2414	0.5107 ± 0.2272	0.5476 ± 0.2424

Tabla 12.3.: Resultados medios de las ejecuciones del CopSoft-SOM.

Nombre	0 % restricciones			10 % restricciones			15 % restricciones			20 % restricciones		
	SOM 2D	SOM 3D	SOM 3D	SOM 2D	SOM 3D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 3D	
Appendicitis	-0.0958 ± 0.0000	-0.0958 ± 0.0000	0.1274 ± 0.3046	0.8317 ± 0.0000	0.3129 ± 0.1132	0.3506 ± 0.0000	0.8004 ± 0.0000	0.8375 ± 0.0000	0.8004 ± 0.0000	0.8375 ± 0.0000	0.8375 ± 0.0000	
Balance	0.1189 ± 0.0000	0.1122 ± 0.0000	0.856 ± 0.001	0.8795 ± 0.1305	0.8635 ± 0.0005	1.0000 ± 0.0000	0.8566 ± 0.0022	1.0000 ± 0.0000	0.8566 ± 0.0022	1.0000 ± 0.0000	1.0000 ± 0.0000	
Banana	-0.0006 ± 0.0000	-0.0006 ± 0.0000	0.8978 ± 0.2992	0.9 ± 0.2999	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	
Breast_cancer	0.009 ± 0.0000	0.009 ± 0.0000	0.6979 ± 0.215	0.3926 ± 0.0924	0.9916 ± 0.0103	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	
Contraceptive	0.1165 ± 0.0000	0.0811 ± 0.0000	0.7097 ± 0.2739	0.9964 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	
Ecoli	0.7944 ± 0.0000	0.7497 ± 0.0000	0.7981 ± 0.0208	0.8361 ± 0.0003	0.8953 ± 0.01	0.8903 ± 0.0006	0.9549 ± 0.0081	0.9432 ± 0.0000	0.9549 ± 0.0081	0.9432 ± 0.0000	0.9432 ± 0.0000	
Glass	0.543 ± 0.0000	0.5153 ± 0.0000	0.5632 ± 0.1203	0.5692 ± 0.026	0.7428 ± 0.0176	0.7247 ± 0.0114	0.8723 ± 0.0207	0.8647 ± 0.0158	0.8723 ± 0.0207	0.8647 ± 0.0158	0.8647 ± 0.0158	
Haberman	-0.0159 ± 0.0000	-0.0159 ± 0.0000	0.3385 ± 0.1143	0.0268 ± 0.0217	0.9674 ± 0.0498	1.0000 ± 0.0000	0.8797 ± 0.2452	1.0000 ± 0.0000	0.8797 ± 0.2452	1.0000 ± 0.0000	1.0000 ± 0.0000	
Hayes_roth	-0.0053 ± 0.0000	-0.0087 ± 0.0000	0.0305 ± 0.0064	0.0525 ± 0.0249	0.0019 ± 0.0059	0.0013 ± 0.0045	0.303 ± 0.1381	0.2359 ± 0.0000	0.303 ± 0.1381	0.2359 ± 0.0000	0.2359 ± 0.0000	
Heart	0.0716 ± 0.0000	0.0716 ± 0.0000	0.0955 ± 0.0687	0.2881 ± 0.0145	0.6933 ± 0.2908	0.9309 ± 0.0609	0.9077 ± 0.2768	1.0000 ± 0.0000	0.9077 ± 0.2768	1.0000 ± 0.0000	1.0000 ± 0.0000	
Ionosphere	-0.0059 ± 0.0000	-0.0059 ± 0.0000	0.0375 ± 0.0511	0.2071 ± 0.1301	0.9673 ± 0.0379	0.9863 ± 0.0112	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	
Iris	0.6562 ± 0.0000	0.6295 ± 0.0000	0.4481 ± 0.0968	0.5794 ± 0.0000	0.6473 ± 0.1412	0.8857 ± 0.0000	0.1927 ± 0.2801	0.9603 ± 0.0000	0.1927 ± 0.2801	0.9603 ± 0.0000	0.9603 ± 0.0000	
Monk2	0.0178 ± 0.0000	0.0191 ± 0.0000	0.8039 ± 0.0731	0.7602 ± 0.1022	0.9781 ± 0.0044	0.9814 ± 0.0015	0.9799 ± 0.0029	0.9817 ± 0.0003	0.9799 ± 0.0029	0.9817 ± 0.0003	0.9817 ± 0.0003	
Newthyroid	0.9192 ± 0.0000	0.8928 ± 0.0000	0.9243 ± 0.0084	0.9214 ± 0.0000	0.7852 ± 0.1873	0.9079 ± 0.0000	0.9541 ± 0.0197	0.9602 ± 0.0000	0.9541 ± 0.0197	0.9602 ± 0.0000	0.9602 ± 0.0000	
Page_blocks	0.0569 ± 0.0000	0.0564 ± 0.0000	0.9807 ± 0.0156	0.9855 ± 0.001	0.994 ± 0.0006	0.9939 ± 0.0006	0.9969 ± 0.002	0.9964 ± 0.0022	0.9969 ± 0.002	0.9964 ± 0.0022	0.9964 ± 0.0022	
Phoneme	0.0016 ± 0.0000	0.0016 ± 0.0000	0.2822 ± 0.3581	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	
Pima	0.0992 ± 0.0000	0.0992 ± 0.0000	0.7388 ± 0.2114	0.9989 ± 0.0021	1.0000 ± 0.0000	0.8005 ± 0.3991	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	
Saheart	0.0051 ± 0.0000	0.0051 ± 0.0000	0.5443 ± 0.2157	0.2819 ± 0.0502	0.6446 ± 0.3745	0.5967 ± 0.3556	0.9366 ± 0.1903	1.0000 ± 0.0000	0.9366 ± 0.1903	1.0000 ± 0.0000	1.0000 ± 0.0000	
Satimage	-0.0008 ± 0.0001	-0.0012 ± 0.0002	0.722 ± 0.2357	0.4495 ± 0.3266	0.9934 ± 0.0074	0.9971 ± 0.0059	0.9988 ± 0.0006	0.9991 ± 0.0003	0.9988 ± 0.0006	0.9991 ± 0.0003	0.9991 ± 0.0003	
Segment	0.0006 ± 0.0000	0.0001 ± 0.0000	0.0117 ± 0.0027	0.0209 ± 0.0006	0.0523 ± 0.0181	0.0584 ± 0.0000	0.9656 ± 0.0437	0.7794 ± 0.0000	0.9656 ± 0.0437	0.7794 ± 0.0000	0.7794 ± 0.0000	
Sonar	0.7146 ± 0.0000	0.6983 ± 0.0000	0.6698 ± 0.0233	0.7815 ± 0.0000	0.0059 ± 0.0041	0.0029 ± 0.0000	0.0019 ± 0.0007	0.0014 ± 0.0000	0.0019 ± 0.0007	0.0014 ± 0.0000	0.0014 ± 0.0000	
Soybean	0.4263 ± 0.0000	0.4199 ± 0.0000	0.4278 ± 0.0186	0.4398 ± 0.0000	0.3683 ± 0.1395	0.5408 ± 0.0234	0.3457 ± 0.0609	0.3653 ± 0.0706	0.3457 ± 0.0609	0.3653 ± 0.0706	0.3653 ± 0.0706	
Spambase	-0.0024 ± 0.0000	-0.0024 ± 0.0000	0.6967 ± 0.4544	0.0003 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	
Spectfheart	-0.0896 ± 0.0000	-0.0896 ± 0.0000	0.3328 ± 0.2359	0.5757 ± 0.0000	0.8842 ± 0.0312	0.9421 ± 0.0204	0.2078 ± 0.2642	0.2832 ± 0.3584	0.2078 ± 0.2642	0.2832 ± 0.3584	0.2832 ± 0.3584	
Tae	0.0135 ± 0.0000	0.0219 ± 0.0000	0.06 ± 0.0163	-0.0086 ± 0.0000	0.0665 ± 0.0234	0.1365 ± 0.0000	0.4418 ± 0.0652	0.3679 ± 0.0000	0.4418 ± 0.0652	0.3679 ± 0.0000	0.3679 ± 0.0000	
Thyroid	-0.0167 ± 0.0000	-0.0167 ± 0.0000	0.8003 ± 0.2651	0.9905 ± 0.0024	0.8692 ± 0.1037	0.8557 ± 0.0903	0.9987 ± 0.0021	0.9989 ± 0.0023	0.9987 ± 0.0021	0.9989 ± 0.0023	0.9989 ± 0.0023	
Titanic	0.0032 ± 0.0000	0.0032 ± 0.0000	0.377 ± 0.3352	0.9877 ± 0.0000	0.6423 ± 0.4489	0.9988 ± 0.0037	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	
Vehicle	0.001 ± 0.0000	0.001 ± 0.0000	0.0886 ± 0.0304	0.0543 ± 0.0205	0.9521 ± 0.0609	0.9735 ± 0.035	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	
Wdbc	0.009 ± 0.0000	0.009 ± 0.0000	0.3667 ± 0.0955	0.6779 ± 0.0908	0.988 ± 0.0032	0.9901 ± 0.0047	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	
Wine	0.649 ± 0.0000	0.661 ± 0.0000	0.6855 ± 0.0225	0.5211 ± 0.0000	0.8348 ± 0.0026	0.8858 ± 0.0000	0.7557 ± 0.065	0.8033 ± 0.0000	0.7557 ± 0.065	0.8033 ± 0.0000	0.8033 ± 0.0000	
Zoo	0.0173 ± 0.0000	-0.0355 ± 0.0000	0.0093 ± 0.0133	-0.0176 ± 0.0215	0.0591 ± 0.0186	0.0366 ± 0.0221	0.1088 ± 0.0313	0.127 ± 0.0443	0.1088 ± 0.0313	0.127 ± 0.0443	0.127 ± 0.0443	
Media	0.1616 ± 0.2879	0.1543 ± 0.2806	0.4878 ± 0.3101	0.5478 ± 0.3643	0.7162 ± 0.345	0.757 ± 0.3459	0.789 ± 0.3154	0.8228 ± 0.3018	0.789 ± 0.3154	0.8228 ± 0.3018	0.8228 ± 0.3018	

Tabla 12.4.: Resultados medios de las ejecuciones del SCCSOM-E.

Nombre	0 % restricciones			10 % restricciones			15 % restricciones			20 % restricciones		
	SOM 2D	SOM 3D	SOM 3D	SOM 2D	SOM 3D	SOM 3D	SOM 2D	SOM 3D	SOM 3D	SOM 2D	SOM 3D	SOM 3D
Appendicitis	0.7262 ± 0.0000	0.8737 ± 0.0000	0.894 ± 0.0202	0.6132 ± 0.3287	0.894 ± 0.0202	0.3253 ± 0.3769	0.7097 ± 0.1322	0.8924 ± 0.1916	0.9607 ± 0.0131	0.8924 ± 0.1916	0.9607 ± 0.0131	0.9607 ± 0.0131
Balance	0.1999 ± 0.0000	0.1999 ± 0.0000	0.7619 ± 0.065	0.6857 ± 0.0889	0.7619 ± 0.065	0.9965 ± 0.0000	0.9968 ± 0.0029	0.9874 ± 0.0334	1.0000 ± 0.0000	0.9874 ± 0.0334	1.0000 ± 0.0000	1.0000 ± 0.0000
Banana	0.995 ± 0.0000	0.995 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
Breast_cancer	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
Contraceptive	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
Ecoli	0.7971 ± 0.0000	0.8254 ± 0.0000	0.8173 ± 0.0000	0.8269 ± 0.0019	0.8173 ± 0.0000	0.8996 ± 0.0086	0.8981 ± 0.0022	0.9666 ± 0.0056	0.9444 ± 0.0000	0.9666 ± 0.0056	0.9444 ± 0.0000	0.9444 ± 0.0000
Glass	0.4956 ± 0.0000	0.5075 ± 0.0000	0.1876 ± 0.0354	0.2399 ± 0.0524	0.1876 ± 0.0354	0.3595 ± 0.0971	0.508 ± 0.102	0.656 ± 0.0403	0.6158 ± 0.0000	0.656 ± 0.0403	0.6158 ± 0.0000	0.6158 ± 0.0000
Haberman	0.9726 ± 0.0000	0.9726 ± 0.0000	0.9878 ± 0.0187	0.9919 ± 0.0163	0.9878 ± 0.0187	0.9863 ± 0.0000	0.9863 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
Hayes_roth	0.0248 ± 0.0000	0.0058 ± 0.0000	0.0754 ± 0.0411	0.0222 ± 0.0251	0.0754 ± 0.0411	0.0448 ± 0.0467	0.0533 ± 0.0273	0.4268 ± 0.0938	0.3741 ± 0.1292	0.4268 ± 0.0938	0.3741 ± 0.1292	0.3741 ± 0.1292
Heart	0.9705 ± 0.0000	0.9705 ± 0.0000	0.9618 ± 0.0174	0.9559 ± 0.0145	0.9618 ± 0.0174	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
Ionosphere	0.9771 ± 0.0000	0.9771 ± 0.0000	0.9863 ± 0.0209	0.9863 ± 0.0209	0.9863 ± 0.0209	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
Iris	0.941 ± 0.0000	0.9601 ± 0.0000	0.786 ± 0.0104	0.7016 ± 0.0875	0.786 ± 0.0104	0.6036 ± 0.1628	0.7896 ± 0.1358	0.7651 ± 0.1838	0.9167 ± 0.0542	0.7651 ± 0.1838	0.9167 ± 0.0542	0.9167 ± 0.0542
Monkz	0.019 ± 0.0000	0.0189 ± 0.0000	0.2871 ± 0.0994	0.2571 ± 0.0369	0.2871 ± 0.0994	0.9785 ± 0.004	0.979 ± 0.0043	0.9799 ± 0.0031	0.9813 ± 0.0004	0.9799 ± 0.0031	0.9813 ± 0.0004	0.9813 ± 0.0004
Newthyroid	0.9467 ± 0.0000	0.9412 ± 0.0000	0.9372 ± 0.0000	0.89 ± 0.0493	0.9372 ± 0.0000	0.9213 ± 0.064	0.9787 ± 0.0000	0.8203 ± 0.0918	0.984 ± 0.0000	0.8203 ± 0.0918	0.984 ± 0.0000	0.984 ± 0.0000
Page_blocks	0.079 ± 0.0019	0.0207 ± 0.0000	0.9849 ± 0.0000	0.9849 ± 0.0004	0.9849 ± 0.0000	0.9873 ± 0.0011	0.9868 ± 0.0006	0.9955 ± 0.001	0.9951 ± 0.0000	0.9955 ± 0.001	0.9951 ± 0.0000	0.9951 ± 0.0000
Phoneme	0.9949 ± 0.0000	0.9949 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
Pima	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
Saheart	0.9826 ± 0.0000	0.9826 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
Satimage	0.0157 ± 0.0001	0.0108 ± 0.0001	0.8681 ± 0.0126	0.8477 ± 0.0349	0.8681 ± 0.0126	0.9994 ± 0.0006	0.9978 ± 0.0041	0.9986 ± 0.0008	0.9993 ± 0.0002	0.9986 ± 0.0008	0.9993 ± 0.0002	0.9993 ± 0.0002
Segment	0.0025 ± 0.0000	0.0016 ± 0.0003	0.0261 ± 0.0097	0.0227 ± 0.0041	0.0261 ± 0.0097	0.0924 ± 0.0349	0.0959 ± 0.0359	0.9471 ± 0.037	0.9759 ± 0.0148	0.9471 ± 0.037	0.9759 ± 0.0148	0.9759 ± 0.0148
Sonar	0.9242 ± 0.0000	0.9242 ± 0.0000	0.1091 ± 0.0000	0.105 ± 0.0123	0.1091 ± 0.0000	0.0006 ± 0.0024	0.0029 ± 0.0000	0.1012 ± 0.2996	0.0014 ± 0.0000	0.1012 ± 0.2996	0.0014 ± 0.0000	0.0014 ± 0.0000
Soybean	0.466 ± 0.0000	0.4744 ± 0.0000	0.4141 ± 0.0000	0.4989 ± 0.0000	0.4141 ± 0.0000	0.4468 ± 0.0509	0.332 ± 0.0947	0.4811 ± 0.0554	0.5018 ± 0.0407	0.4811 ± 0.0554	0.5018 ± 0.0407	0.5018 ± 0.0407
Spambase	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
Spectfheart	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
Tae	0.1404 ± 0.0000	0.1306 ± 0.0000	0.2775 ± 0.0117	0.1906 ± 0.0236	0.2775 ± 0.0117	0.2985 ± 0.1168	0.3605 ± 0.1775	0.6395 ± 0.1222	0.6221 ± 0.176	0.6395 ± 0.1222	0.6221 ± 0.176	0.6221 ± 0.176
Thyroid	0.0015 ± 0.0000	0.0004 ± 0.0025	0.9892 ± 0.0004	0.9892 ± 0.0004	0.9898 ± 0.0014	0.9924 ± 0.0014	0.9931 ± 0.0016	0.999 ± 0.0006	0.9994 ± 0.0002	0.999 ± 0.0006	0.9994 ± 0.0002	0.9994 ± 0.0002
Titanic	0.9877 ± 0.0000	0.9877 ± 0.0000	0.9939 ± 0.0000	0.9939 ± 0.0000	0.9939 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
Vehicle	0.9751 ± 0.0000	0.9751 ± 0.0000	0.982 ± 0.005	0.9068 ± 0.0874	0.982 ± 0.005	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
Wdbc	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
Wine	0.7834 ± 0.0000	0.771 ± 0.0000	0.7385 ± 0.0000	0.6893 ± 0.1011	0.7385 ± 0.0000	0.9117 ± 0.0711	0.9472 ± 0.0000	0.708 ± 0.1498	0.9832 ± 0.0000	0.708 ± 0.1498	0.9832 ± 0.0000	0.9832 ± 0.0000
Zoo	0.0 ± 0.0000	-0.009 ± 0.0000	-0.0009 ± 0.0279	0.0029 ± 0.0118	-0.0009 ± 0.0279	0.9999 ± 0.0517	0.0957 ± 0.063	0.226 ± 0.083	0.269 ± 0.1561	0.226 ± 0.083	0.269 ± 0.1561	0.269 ± 0.1561
Media	0.6587 ± 0.4099	0.6617 ± 0.4167	0.744 ± 0.352	0.7227 ± 0.3533	0.744 ± 0.352	0.7724 ± 0.3494	0.7972 ± 0.3355	0.8578 ± 0.2419	0.875 ± 0.2534	0.8578 ± 0.2419	0.875 ± 0.2534	0.875 ± 0.2534

Tabla 12.5.: Resultados medios de las ejecuciones del SCCSOM-EAL-SL.

12. Resultados experimentales.

A partir de las tablas, y de la información de los resultados medios que se han recopilado en cada época. se han podido obtener gráficas. Estas calculan el ARI para el agrupamiento que resulta en cada una de las 30 primeras épocas para cada algoritmo, dando así una idea de cómo es su convergencia.

En estas gráficas se observa el valor medio de los algoritmos para todos los conjuntos de datos en cada época, marcando con distintos colores las gráficas de la topología de la red, y con distintos tipos de línea el conjunto de restricciones usado.

En la [Figura 12.1](#) y [Figura 12.3](#) se observa que los algoritmos SCCSOM y SoftSom, convergen en unas 4 o 5 épocas a su mejor solución y que el aumento de restricciones en estos dos casos provoca una mejora incremental en las soluciones. En cambio en la [Figura 12.2](#), se observa como el método converge en un par de épocas, siguiendo todas las ejecuciones que llevan restricciones la misma gráfica, sin que el porcentaje de restricciones afecte, siempre que sea mayor que el 10 %.

En los métodos de “ensemble”, [Figura 12.4](#) y [Figura 12.5](#), se observa una convergencia mucho más rápida que en los otros métodos y también un ajuste incremental en la calidad de las soluciones según su porcentaje de restricciones. Lo que si se aprecia es la diferencia entre los dos métodos de ordenación para los conjuntos de menos restricciones, una diferencia que se mantiene durante todas las épocas.

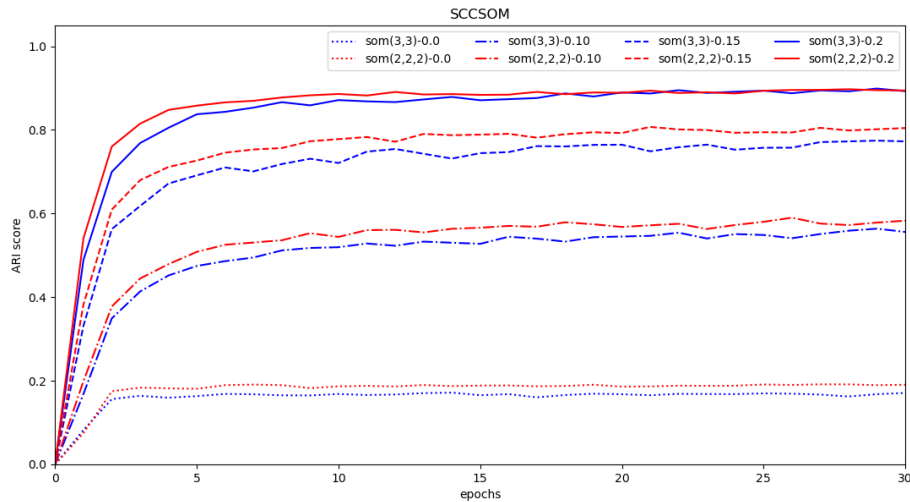


Figura 12.1.: Gráfica comparativa SCCSOM.

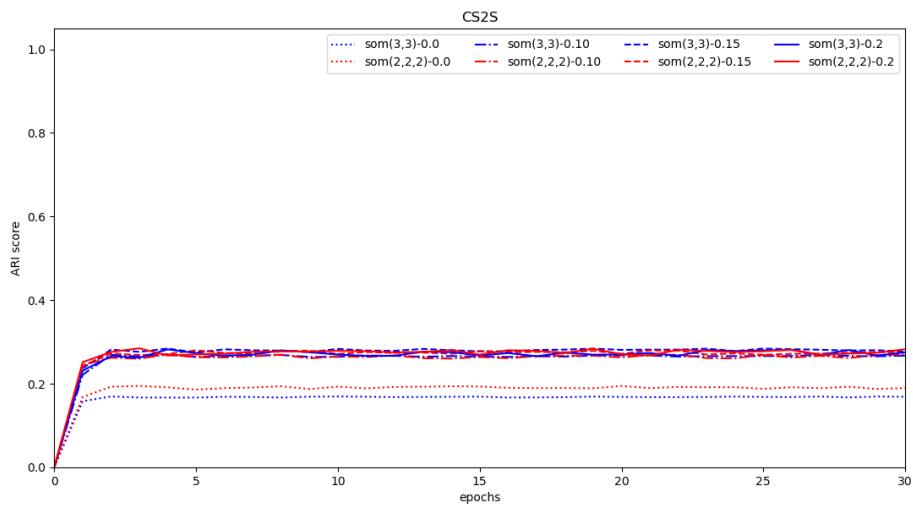


Figura 12.2.: Gráfica comparativa CS2S.

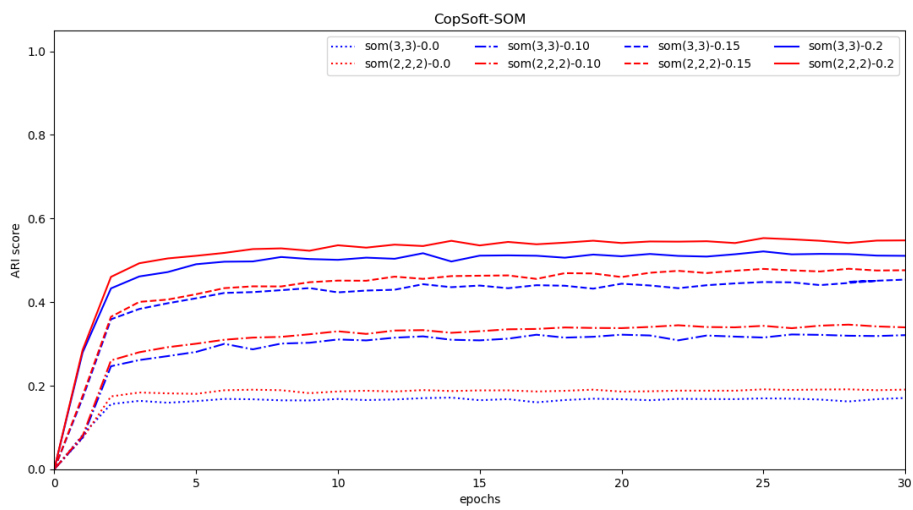


Figura 12.3.: Gráfica comparativa CopSoft-SOM.

12. Resultados experimentales.

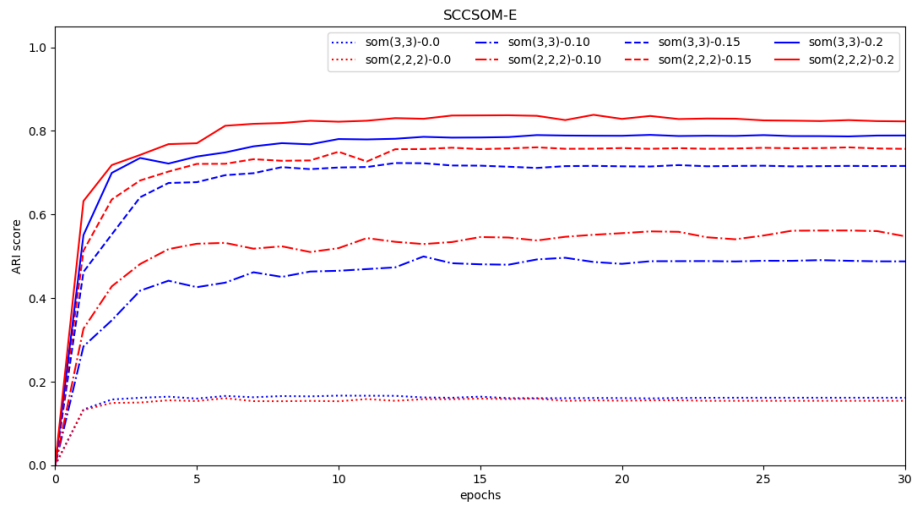


Figura 12.4.: Gráfica comparativa SCCSOM-E.

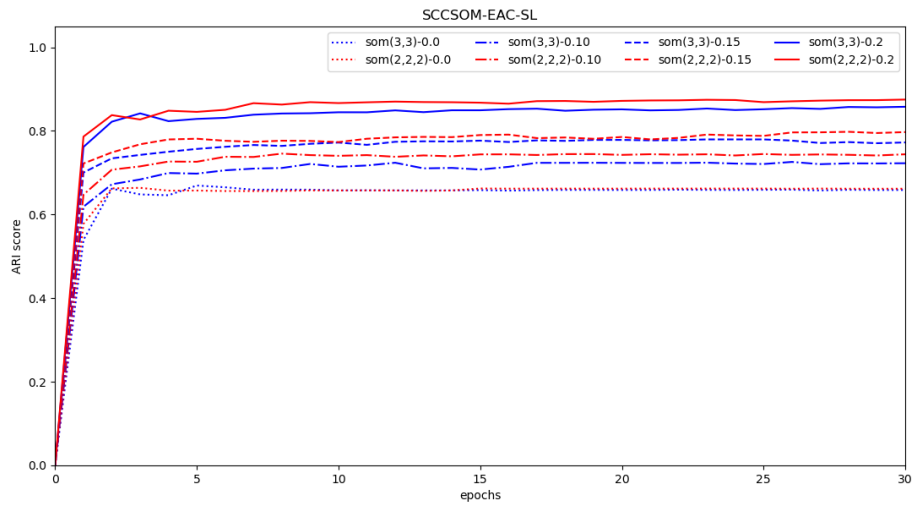


Figura 12.5.: Gráfica comparativa SCCSOM-EAC-SL.

13. Conclusión y Trabajo Futuro

13.1. Conclusión

En este trabajo se ha propuesto el algoritmo SCCSOM, un método de agrupamiento con restricciones “soft” que utiliza un mapa auto-organizativo. Como ya se ha dicho en la introducción, el enfoque de usar restricciones “soft” no había sido tratado para este problema usando mapas auto-organizativos, sólo existían propuestas con restricciones “hard”. Por lo que además de realizar nuestra propuesta se han tenido que adaptar las ya existentes al caso de restricciones “soft”, como son el SC2GS y el Cop-SOM.

Tras la experimentación con los tres algoritmos se ha demostrado la idoneidad de mi propuesta, ya que para los conjuntos de datos probados, SCCSOM mejora por mucha diferencia a los otros dos algoritmos conforme aumentan el número de restricciones. Esto ha venido a demostrar tanto la idoneidad del algoritmo SCCSOM para el problema, puesto que presenta muy buenos resultados, como su utilidad ya que es un enfoque dado al problema del agrupamiento de restricciones que nunca se había hecho con mapas auto-organizativos y porque la adaptación de los algoritmos que ya habían sido implementados para un problema parecido no son capaces de conseguir alcanzar buenos resultados.

Además, se han implementado dos métodos de ensemble, SCCSOM-E y SCCSOM-EAC-SL, que tienen como base el algoritmo propuesto SCCSOM para clustering con restricciones “soft”. Tras la experimentación se ha demostrado que los mapas auto-organizativos ofrecen un muy buen resultado utilizándose como última capa de estos sistemas de aprendizaje múltiple, abriendo la posibilidad a la implementación de mapas auto-organizativos de varias capas de profundidad o a sistemas más complejos que cuenten con un SOM de capa final.

cabe destacar también que el objetivo marcado de profundizar en la base matemática se ha alcanzado, explicando con detalle los procesos de Markov y la cuantificación vectorial, además de demostrar el fenómeno de auto-organización. Junto a esto se ha conseguido dar a un problema actual y clásico de la informática un formalismo matemático accesible y entendible para cualquier persona con conocimientos intermedios en Matemáticas.

Por tanto podemos concluir con que el trabajo ha conseguido los objetivos que se buscaban, demostrando la idoneidad del uso de restricciones y mapas auto-organizativos para el problema del agrupamiento.

13.2. Trabajo Futuro

De cara a seguir en el futuro desarrollando el tema abordado, se proponen dos líneas de investigación, una más matemática y otra más de desarrollo e implementación de algoritmos.

13. Conclusión y Trabajo Futuro

La parte más matemática podría consistir en aplicar técnicas estadísticas más sofisticadas, como pueden ser los test estadísticos o los modelos bayesianos, a los resultados de la experimentación con tal de sacar más conclusiones de las que se han podido extraer en este trabajo. Otra posible línea de investigación matemática podría ser el diseño de topologías más complejas para los mapas o el diseño de distancias en el mapa donde influyan los pesos de las neuronas, las épocas y sobre todo las restricciones, y no sólo la topología del mapa.

Una línea de investigación más enfocada en el desarrollo e implementación de algoritmos podría centrarse en el estudio de distintas variantes del SOM, como puede ser el GSOM, y siguiendo la estrategia que se ha hecho en este trabajo para el SOM básico, adaptar el problema del clustering con restricciones a estos nuevos tipos de mapas auto-organizativos. Otra posible línea de investigación podría ser adaptar la librería de SOM para clustering con restricciones para que sus métodos más pesados como el entrenamiento de la red pudiera ser ejecutado en paralelo con distintas hebras, función de la que todavía no dispone. Además se podría investigar una hibridación del mapa auto-organizativo con un algoritmo genético que ayudara a mejorar aún más la búsqueda de la configuración óptima de clusters.

Finalmente extendiendo la última parte de la sección anterior, una línea de investigación interesante sería la investigación acerca de la posibilidad de crear mapas auto-organizativos de varias capas y las posibles aplicaciones de estos mapas auto-organizativos que formarían redes multicapa de neuronas, asemejándose a las redes densas existentes.

A. Resultados experimentales completos.

A continuación se presentan los desgloses de ejecuciones de los resultados medios presentados en las tablas del [Capítulo 12](#), para los algoritmos [SCCSOM](#), [CS2S](#), [CopSoft-SOM](#), [SCCSOM-E](#) y [SCCSOM-EAL-SL](#).

Tabla A.1.: Resultados de las ejecuciones del SCCSOM.

Nombre	0% restricciones		10% restricciones		15% restricciones		20% restricciones	
	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D
Appendicitis	0.5164 ± 0.0247	0.4975 ± 0.0343	0.2604 ± 0.1938	0.2674 ± 0.2882	0.3698 ± 0.2437	0.4776 ± 0.1948	0.6554 ± 0.2661	0.8308 ± 0.135
Ejecución 1	0.5153	0.5256	0.5146	0.6703	0.4318	0.2959	1.0	0.4923
Ejecución 2	0.4746	0.5256	0.2133	0.4062	-0.0768	0.4873	0.7806	0.7806
Ejecución 3	0.5153	0.4862	0.162	-0.0101	0.4466	0.6235	0.8713	0.8713
Ejecución 4	0.5153	0.4199	0.4169	0.0033	0.6041	0.0582	0.6968	0.8004
Ejecución 5	0.5459	0.4966	0.3431	0.6703	0.5348	0.3094	0.0455	0.9125
Ejecución 6	0.5153	0.5256	0.5432	-0.0269	0.203	0.573	0.5081	0.9571
Ejecución 7	0.5459	0.5256	0.0328	0.0611	0.6287	0.4619	0.5962	0.9571
Ejecución 8	0.4746	0.4583	0.0266	0.1448	0.1362	0.6989	0.6365	0.7422
Ejecución 9	0.5459	0.5256	-0.0072	0.6703	0.1191	0.7169	0.9563	0.8375
Ejecución 10	0.5153	0.4862	0.3585	0.845	0.6703	0.5508	0.4623	0.9571
Balance	0.0607 ± 0.0177	0.0578 ± 0.0312	0.7973 ± 0.1238	0.8282 ± 0.0482	0.9874 ± 0.0242	0.9951 ± 0.0028	0.9771 ± 0.0362	0.9595 ± 0.0532
Ejecución 1	0.06	0.0168	0.862	0.8121	0.986	0.993	0.9497	1.0
Ejecución 2	0.0628	0.0903	0.6356	0.8587	0.993	0.993	1.0	0.9221
Ejecución 3	0.0618	0.094	0.8794	0.7943	0.993	0.993	1.0	0.9692
Ejecución 4	0.0481	0.0614	0.4927	0.8336	1.0	0.993	1.0	0.8602
Ejecución 5	0.0604	0.0723	0.8499	0.8212	0.9965	0.993	1.0	1.0
Ejecución 6	0.0746	0.0112	0.8087	0.7927	0.993	1.0	0.9895	0.9759
Ejecución 7	0.0261	0.0326	0.8126	0.738	0.9159	0.993	0.9434	1.0
Ejecución 8	0.1002	0.094	0.862	0.8279	1.0	0.9965	1.0	0.8675
Ejecución 9	0.0559	0.0779	0.9047	0.9188	0.9965	0.9965	1.0	1.0
Ejecución 10	0.0572	0.0273	0.8651	0.8848	1.0	1.0	0.8883	1.0
Banana	0.003 ± 0.0036	0.0437 ± 0.0306	0.8983 ± 0.2993	0.899 ± 0.2996	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	0.0083	0.009	1.0	0.9975	1.0	1.0	1.0	1.0
Ejecución 2	0.0037	0.0698	0.9975	1.0	1.0	1.0	1.0	1.0
Ejecución 3	0.0012	0.0842	0.9975	0.9975	1.0	1.0	1.0	1.0
Ejecución 4	0.003	0.0137	1.0	0.9975	1.0	1.0	1.0	1.0
Ejecución 5	0.0024	0.0203	0.9975	1.0	1.0	1.0	1.0	1.0
Ejecución 6	0.0004	0.0755	0.9975	0.9975	1.0	1.0	1.0	1.0
Ejecución 7	0.0023	0.0185	0.0003	1.0	1.0	1.0	1.0	1.0
Ejecución 8	-0.005	0.0247	0.9975	1.0	1.0	1.0	1.0	1.0
Ejecución 9	0.0057	0.0304	0.9975	0.0003	1.0	1.0	1.0	1.0
Ejecución 10	0.0075	0.0903	0.9975	1.0	1.0	1.0	1.0	1.0
Breast_cancer	0.2824 ± 0.0212	0.3576 ± 0.0122	0.9581 ± 0.0127	0.9588 ± 0.0121	0.9986 ± 0.0028	0.9944 ± 0.0028	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	0.2537	0.3631	0.9512	0.9719	0.9929	0.9929	1.0	1.0
Ejecución 2	0.3089	0.3538	0.9374	0.9374	1.0	0.9929	1.0	1.0
Ejecución 3	0.2872	0.3447	0.9512	0.965	1.0	0.9929	1.0	1.0
Ejecución 4	0.2787	0.3724	0.9512	0.958	1.0	0.9929	1.0	1.0
Ejecución 5	0.2456	0.3447	0.9789	0.9374	1.0	1.0	1.0	1.0
Ejecución 6	0.3045	0.3771	0.9581	0.9511	1.0	0.9929	1.0	1.0
Ejecución 7	0.2703	0.3401	0.9443	0.965	1.0	0.9929	1.0	1.0
Ejecución 8	0.3045	0.3492	0.9719	0.965	1.0	0.9929	1.0	1.0
Ejecución 9	0.3002	0.3681	0.9719	0.965	0.9929	1.0	1.0	1.0
Ejecución 10	0.2703	0.3631	0.965	0.9719	1.0	0.9929	1.0	1.0
Contraceptive	0.0118 ± 0.001	0.022 ± 0.0011	0.9924 ± 0.0045	0.991 ± 0.0015	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	0.0102	0.0236	0.993	0.9915	1.0	1.0	1.0	1.0
Ejecución 2	0.0128	0.0197	0.9947	0.9896	1.0	1.0	1.0	1.0
Ejecución 3	0.0116	0.0219	0.9913	0.9913	1.0	1.0	1.0	1.0
Ejecución 4	0.0117	0.023	0.9896	0.9896	1.0	1.0	1.0	1.0
Ejecución 5	0.014	0.0221	0.9966	0.9913	1.0	1.0	1.0	1.0
Ejecución 6	0.0115	0.0221	0.9913	0.9896	1.0	1.0	1.0	1.0
Ejecución 7	0.0112	0.0211	0.9932	0.9913	1.0	1.0	1.0	1.0
Ejecución 8	0.0109	0.0233	0.9966	0.9896	1.0	1.0	1.0	1.0
Ejecución 9	0.0122	0.022	0.9808	0.9947	1.0	1.0	1.0	1.0
Ejecución 10	0.0118	0.0208	0.9966	0.9913	1.0	1.0	1.0	1.0
Ecoli	0.2111 ± 0.0083	0.2989 ± 0.0142	0.2101 ± 0.0851	0.232 ± 0.053	0.79 ± 0.0343	0.7207 ± 0.155	0.9134 ± 0.0355	0.9405 ± 0.0157
Ejecución 1	0.2147	0.3121	0.187	0.2515	0.818	0.4365	0.9176	0.9354
Ejecución 2	0.2095	0.3088	0.1389	0.2679	0.7777	0.7588	0.8868	0.9457
Ejecución 3	0.2166	0.2845	0.1501	0.2906	0.8175	0.8564	0.9441	0.9339
Ejecución 4	0.2126	0.2967	0.4203	0.1734	0.7872	0.8277	0.9255	0.9487
Ejecución 5	0.2043	0.2806	0.1725	0.2626	0.817	0.8297	0.9354	0.9233
Ejecución 6	0.2253	0.2945	0.123	0.1483	0.7879	0.7798	0.9295	0.909
Ejecución 7	0.2019	0.2871	0.1492	0.1718	0.7034	0.5532	0.8278	0.9461
Ejecución 8	0.1952	0.3044	0.2346	0.2548	0.8158	0.8553	0.8846	0.9679
Ejecución 9	0.2181	0.3297	0.2532	0.3081	0.764	0.4846	0.9278	0.9392
Ejecución 10	0.2122	0.2907	0.2726	0.1906	0.8118	0.8254	0.9546	0.9552
Glass	0.213 ± 0.0202	0.2307 ± 0.0171	0.1476 ± 0.0397	0.1485 ± 0.0158	0.2831 ± 0.1516	0.3482 ± 0.1317	0.7827 ± 0.0622	0.7478 ± 0.0923
Ejecución 1	0.2121	0.2587	0.1297	0.1618	0.3721	0.2402	0.8129	0.6264
Ejecución 2	0.2039	0.2137	0.2192	0.1265	0.2543	0.5614	0.8433	0.7389

Continúa en la siguiente página

A. Resultados experimentales completos.

Tabla A.1 – continúa de la página anterior								
Nombre	0 % restricciones		10 % restricciones		15 % restricciones		20 % restricciones	
	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D
Ejecución 3	0.1803	0.2301	0.1047	0.1614	0.1808	0.2991	0.7161	0.9148
Ejecución 4	0.2361	0.2063	0.1243	0.1795	0.4011	0.2725	0.819	0.8537
Ejecución 5	0.2365	0.2524	0.0937	0.1404	0.188	0.6116	0.6438	0.6222
Ejecución 6	0.1934	0.2384	0.1214	0.1356	0.6204	0.3459	0.7394	0.6915
Ejecución 7	0.2154	0.2178	0.1618	0.1348	0.1594	0.2982	0.7969	0.7378
Ejecución 8	0.2247	0.2159	0.1388	0.135	0.1324	0.2711	0.7879	0.8142
Ejecución 9	0.188	0.248	0.2026	0.1566	0.3762	0.1821	0.8044	0.8017
Ejecución 10	0.2391	0.2258	0.1798	0.1537	0.1375	0.4002	0.8636	0.6772
Haberman	-0.0136 ± 0.0109	-0.0247 ± 0.0106	0.166 ± 0.1323	0.3044 ± 0.0944	0.9182 ± 0.2453	0.9825 ± 0.0525	0.8817 ± 0.2268	0.8678 ± 0.295
Ejecución 1	-0.0086	-0.0277	0.0493	0.3065	1.0	1.0	0.4109	1.0
Ejecución 2	-0.007	-0.0123	0.3317	0.4511	1.0	1.0	0.9589	1.0
Ejecución 3	-0.0068	-0.0343	0.0263	0.3202	1.0	1.0	0.4471	1.0
Ejecución 4	-0.0165	-0.0107	0.1085	0.1393	0.1823	0.825	1.0	0.0463
Ejecución 5	-0.0145	-0.0107	0.0668	0.3872	1.0	1.0	1.0	0.6321
Ejecución 6	-0.0127	-0.0329	0.4266	0.1557	1.0	1.0	1.0	1.0
Ejecución 7	-0.0437	-0.0282	0.0405	0.3883	1.0	1.0	1.0	1.0
Ejecución 8	-0.0165	-0.0369	0.1382	0.2685	1.0	1.0	1.0	1.0
Ejecución 9	-0.003	-0.0155	0.2955	0.3497	1.0	1.0	1.0	1.0
Ejecución 10	-0.007	-0.0372	0.1764	0.2775	1.0	1.0	1.0	1.0
Hayes_roth	0.0802 ± 0.0223	0.0224 ± 0.0285	0.0438 ± 0.0416	0.0466 ± 0.0481	0.219 ± 0.1201	0.186 ± 0.1017	0.6017 ± 0.1228	0.6018 ± 0.112
Ejecución 1	0.1334	0.0533	0.0117	0.0198	0.1122	0.1276	0.5978	0.5901
Ejecución 2	0.1007	0.0162	0.0698	-0.0033	0.2032	0.4134	0.6663	0.5053
Ejecución 3	0.0902	-0.0131	0.0145	0.0513	0.4254	0.1027	0.4665	0.5544
Ejecución 4	0.0603	0.0476	0.1175	0.0327	0.1904	0.0895	0.7377	0.5853
Ejecución 5	0.0732	0.0043	0.0015	0.0016	0.2283	0.3064	0.8061	0.809
Ejecución 6	0.0693	0.0193	0.1002	0.0736	0.1153	0.2246	0.7055	0.7509
Ejecución 7	0.0692	0.0706	0.0813	0.1401	0.3366	0.1573	0.3854	0.57
Ejecución 8	0.0778	0.0442	0.0171	0.122	0.0865	0.2289	0.517	0.5753
Ejecución 9	0.0784	-0.0166	0.0026	-0.0003	0.3967	0.118	0.5449	0.6781
Ejecución 10	0.0495	-0.0016	0.0216	0.0281	0.0954	0.0919	0.5898	0.3997
Heart	0.0033 ± 0.0034	0.0085 ± 0.0036	0.0806 ± 0.082	0.2005 ± 0.1338	0.6773 ± 0.2835	0.8739 ± 0.2902	0.9219 ± 0.2343	0.9141 ± 0.2576
Ejecución 1	0.0034	0.0116	-0.004	-0.0037	0.3309	0.9852	1.0	1.0
Ejecución 2	0.0005	0.0074	0.0276	0.1232	0.9852	0.0043	1.0	1.0
Ejecución 3	-0.0008	0.0131	-0.0031	0.1568	0.5037	0.9414	1.0	1.0
Ejecución 4	0.0033	0.0028	0.0981	0.258	0.4523	0.9852	1.0	1.0
Ejecución 5	0.0049	0.0094	0.2076	0.2148	0.9705	0.9705	0.2191	1.0
Ejecución 6	0.0009	0.0134	0.2279	0.1746	0.3753	0.9852	1.0	1.0
Ejecución 7	-0.0006	0.0094	0.1395	0.4933	0.927	0.9559	1.0	1.0
Ejecución 8	0.0071	0.0096	0.0406	0.357	0.3308	0.9852	1.0	0.1413
Ejecución 9	0.0105	0.0041	0.075	0.1335	0.927	0.9559	1.0	1.0
Ejecución 10	0.0036	0.0042	-0.0034	0.0976	0.9705	0.9705	1.0	1.0
Ionosphere	0.0106 ± 0.0894	0.0019 ± 0.0192	0.356 ± 0.1823	0.5758 ± 0.12	0.8829 ± 0.2936	0.9888 ± 0.0336	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	-0.0104	0.0438	0.4006	0.7146	1.0	1.0	1.0	1.0
Ejecución 2	-0.0329	-0.0077	0.0249	0.6485	1.0	1.0	1.0	1.0
Ejecución 3	-0.017	0.0241	0.1861	0.5374	0.9658	0.888	1.0	1.0
Ejecución 4	-0.0074	-0.0211	0.5643	0.3874	1.0	1.0	1.0	1.0
Ejecución 5	-0.0083	0.0006	0.4347	0.7051	1.0	1.0	1.0	1.0
Ejecución 6	0.2768	0.0115	0.4528	0.5239	0.009	1.0	1.0	1.0
Ejecución 7	-0.0074	-0.0019	0.5173	0.6768	0.9771	1.0	1.0	1.0
Ejecución 8	-0.0153	-0.0136	0.5717	0.6303	0.8771	1.0	1.0	1.0
Ejecución 9	-0.033	0.0029	0.2871	0.3525	1.0	1.0	1.0	1.0
Ejecución 10	-0.039	-0.0194	0.1205	0.5819	1.0	1.0	1.0	1.0
Iris	0.6319 ± 0.0697	0.7542 ± 0.0716	0.6493 ± 0.1075	0.6807 ± 0.1283	0.7834 ± 0.1402	0.8374 ± 0.0994	0.8977 ± 0.0914	0.9045 ± 0.0633
Ejecución 1	0.7196	0.688	0.772	0.7323	0.8441	0.9039	0.8664	0.8664
Ejecución 2	0.5496	0.8343	0.6545	0.7432	0.941	0.7511	0.941	0.9603
Ejecución 3	0.5587	0.8341	0.7417	0.5455	0.7145	0.6659	0.7196	0.9603
Ejecución 4	0.7196	0.6643	0.5517	0.6598	0.7862	0.941	1.0	0.9603
Ejecución 5	0.5923	0.7145	0.6521	0.7245	0.7388	0.9039	0.9603	0.9406
Ejecución 6	0.7196	0.8184	0.5087	0.3803	0.941	0.8316	0.7251	0.9039
Ejecución 7	0.7028	0.818	0.6653	0.7297	0.4686	0.9603	0.941	0.8188
Ejecución 8	0.6011	0.8178	0.8508	0.7424	0.941	0.9603	0.9039	0.772
Ejecución 9	0.58	0.676	0.5748	0.818	0.6712	0.8611	0.941	0.9603
Ejecución 10	0.5762	0.6765	0.5217	0.8305	0.788	0.7664	0.941	0.9026
Monk2	0.0965 ± 0.0423	0.1144 ± 0.0368	0.5273 ± 0.1082	0.5035 ± 0.0953	0.9805 ± 0.0027	0.9817 ± 0.0017	0.9533 ± 0.0823	0.9818 ± 0.0005
Ejecución 1	0.0377	0.1126	0.4956	0.4047	0.9735	0.9816	0.9821	0.9821
Ejecución 2	0.1732	0.1065	0.4443	0.4813	0.9772	0.9821	0.7064	0.9821
Ejecución 3	0.0993	0.0594	0.4503	0.3184	0.9816	0.9821	0.9816	0.9816
Ejecución 4	0.1602	0.1357	0.723	0.6704	0.9816	0.9827	0.9816	0.9811
Ejecución 5	0.0868	0.1946	0.4725	0.5179	0.9821	0.9827	0.9821	0.9811
Ejecución 6	0.1108	0.0829	0.5969	0.5385	0.9816	0.9827	0.9827	0.9821
Ejecución 7	0.0704	0.1496	0.6874	0.4981	0.9821	0.9827	0.9816	0.9821
Ejecución 8	0.066	0.0887	0.35	0.5638	0.9816	0.9768	0.9816	0.9827
Ejecución 9	0.0485	0.1232	0.5309	0.6005	0.9816	0.9821	0.9719	0.9816
Ejecución 10	0.1125	0.0906	0.522	0.441	0.9821	0.9821	0.9811	0.9816
Newthyroid	0.5108 ± 0.1137	0.4953 ± 0.0325	0.2706 ± 0.1131	0.4055 ± 0.1511	0.8551 ± 0.0413	0.831 ± 0.0544	0.9215 ± 0.116	0.9605 ± 0.0253
Ejecución 1	0.6522	0.4818	0.2754	0.638	0.848	0.7823	0.9728	0.984
Ejecución 2	0.5997	0.4801	0.342	0.4491	0.8443	0.8789	0.9728	0.984
Ejecución 3	0.5437	0.5644	0.4009	0.316	0.8173	0.8459	0.9581	0.9137
Ejecución 4	0.5011	0.5168	0.2694	0.3032	0.9317	0.8137	0.9479	0.9311
Ejecución 5	0.5588	0.474	0.1338	0.4717	0.906	0.9105	0.9675	0.9783
Ejecución 6	0.4553	0.4488	0.024	0.5829	0.8279	0.8633	0.5787	0.9678
Ejecución 7	0.5156	0.4892	0.3868	0.5111	0.8579	0.8466	0.9536	0.9728
Ejecución 8	0.4687	0.5324	0.2321	0.4224	0.842	0.7708	0.9783	0.963
Ejecución 9	0.2177	0.4661	0.3769	0.2304	0.7853	0.7243	0.9071	0.9261
Ejecución 10	0.5952	0.4995	0.2649	0.1301	0.80	0.8734	0.9783	0.984
Page_blocks	-0.009 ± 0.0042	-0.0253 ± 0.0022	0.9851 ± 0.0006	0.9856 ± 0.0005	0.9907 ± 0.0033	0.9904 ± 0.0031	0.9968 ± 0.0023	0.9968 ± 0.0017
Ejecución 1	-0.0087	-0.0265	0.9851	0.9867	0.9958	0.995	0.9968	0.997
Ejecución 2	-0.0073	-0.0271	0.9847	0.9854	0.9888	0.9958	0.9975	0.9975
Ejecución 3	-0.0206	-0.0269	0.9847	0.9851	0.9951	0.9886	0.9976	0.997
Ejecución 4	-0.0073	-0.027	0.9863	0.9854	0.9949	0.995	0.9977	0.9917
Ejecución 5	-0.0066	-0.0249	0.9843	0.9857	0.9908	0.9884	0.9974	0.9975

Continúa en la siguiente página

Tabla A.1 – continúa de la página anterior								
Nombre	0% restricciones		10% restricciones		15% restricciones		20% restricciones	
	SOM \pm 2D	SOM \pm 3D	SOM \pm 2D	SOM \pm 3D	SOM \pm 2D	SOM \pm 3D	SOM \pm 2D	SOM \pm 3D
Ejecución 6	-0.0044	-0.025	0.9849	0.986	0.9897	0.9896	0.9981	0.9977
Ejecución 7	-0.0077	-0.0231	0.9862	0.9853	0.9907	0.9867	0.9974	0.9972
Ejecución 8	-0.0074	-0.0244	0.985	0.9856	0.986	0.9875	0.998	0.9977
Ejecución 9	-0.01	-0.0282	0.9852	0.9859	0.9882	0.9946	0.9899	0.9973
Ejecución 10	-0.0095	-0.0205	0.9848	0.9848	0.9873	0.9887	0.9973	0.9972
Phoneme	0.2188 \pm 0.0104	0.2613 \pm 0.0097	0.8982 \pm 0.2977	0.8977 \pm 0.2975	1.0 \pm 0.0	1.0 \pm 0.0	1.0 \pm 0.0	1.0 \pm 0.0
Ejecución 1	0.2122	0.2468	0.9949	0.9949	1.0	1.0	1.0	1.0
Ejecución 2	0.2261	0.2634	0.9949	1.0	1.0	1.0	1.0	1.0
Ejecución 3	0.2067	0.2702	0.9949	1.0	1.0	1.0	1.0	1.0
Ejecución 4	0.236	0.2462	1.0	0.9975	1.0	1.0	1.0	1.0
Ejecución 5	0.2198	0.2674	0.9975	0.9949	1.0	1.0	1.0	1.0
Ejecución 6	0.2035	0.2568	0.0052	0.9949	1.0	1.0	1.0	1.0
Ejecución 7	0.2191	0.2736	1.0	0.0052	1.0	1.0	1.0	1.0
Ejecución 8	0.2088	0.2513	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 9	0.2324	0.2696	0.9949	0.9949	1.0	1.0	1.0	1.0
Ejecución 10	0.223	0.2678	1.0	0.9949	1.0	1.0	1.0	1.0
Pima	0.0473 \pm 0.0045	-0.0009 \pm 0.0004	0.9892 \pm 0.0274	0.9974 \pm 0.0026	0.9958 \pm 0.0021	0.9968 \pm 0.0026	1.0 \pm 0.0	1.0 \pm 0.0
Ejecución 1	0.0567	-0.0014	1.0	1.0	0.9947	1.0	1.0	1.0
Ejecución 2	0.0442	-0.0008	1.0	0.9947	0.9947	1.0	1.0	1.0
Ejecución 3	0.0441	-0.0005	1.0	0.9947	0.9947	1.0	1.0	1.0
Ejecución 4	0.0517	-0.0016	1.0	0.9947	0.9947	1.0	1.0	1.0
Ejecución 5	0.0461	-0.0012	0.9947	1.0	0.9947	0.9947	1.0	1.0
Ejecución 6	0.0518	-0.0007	0.9947	1.0	1.0	0.9947	1.0	1.0
Ejecución 7	0.0481	-0.0011	0.9947	0.9947	0.9947	0.9947	1.0	1.0
Ejecución 8	0.0462	-0.0001	1.0	0.9947	0.9947	0.9947	1.0	1.0
Ejecución 9	0.0419	-0.0004	1.0	1.0	0.9947	0.9947	1.0	1.0
Ejecución 10	0.0426	-0.0008	0.9073	1.0	0.9947	1.0	1.0	1.0
Saheart	-0.0035 \pm 0.0165	-0.0079 \pm 0.0081	0.7163 \pm 0.2285	0.8227 \pm 0.045	0.5071 \pm 0.4222	0.9835 \pm 0.0099	0.944 \pm 0.1679	1.0 \pm 0.0
Ejecución 1	0.0125	-0.0119	0.8332	0.8651	0.5061	0.9913	1.0	1.0
Ejecución 2	0.0179	-0.0095	0.8492	0.865	0.0116	0.9913	1.0	1.0
Ejecución 3	0.0015	0.0087	0.8491	0.7864	0.1672	0.9653	1.0	1.0
Ejecución 4	0.003	0.0008	0.8412	0.7864	0.0116	0.9913	1.0	1.0
Ejecución 5	-0.0303	-0.0096	0.7009	0.8253	0.0116	0.9826	0.4404	1.0
Ejecución 6	-0.0264	-0.0146	0.7388	0.8571	0.9653	0.9826	1.0	1.0
Ejecución 7	0.0074	-0.0186	0.8412	0.8174	0.415	0.9739	1.0	1.0
Ejecución 8	-0.0011	-0.0112	0.8572	0.7183	0.9913	0.9739	1.0	1.0
Ejecución 9	-0.0256	-0.0138	0.5682	0.8491	0.9913	0.9826	1.0	1.0
Ejecución 10	0.0058	0.0005	0.0841	0.8572	1.0	1.0	1.0	1.0
Satimage	0.3654 \pm 0.0036	0.4634 \pm 0.013	0.9747 \pm 0.0079	0.9746 \pm 0.0134	0.9989 \pm 0.0008	0.9965 \pm 0.0056	0.9985 \pm 0.0008	0.9992 \pm 0.0005
Ejecución 1	0.367	0.4846	0.9758	0.94	0.9993	0.9996	0.9996	0.9993
Ejecución 2	0.3651	0.4822	0.9588	0.9788	0.9981	0.9992	0.9981	0.9996
Ejecución 3	0.365	0.4618	0.9697	0.9844	0.9996	0.9993	0.9982	0.9986
Ejecución 4	0.3619	0.4713	0.9812	0.987	0.9974	0.9996	0.9986	0.9985
Ejecución 5	0.3679	0.458	0.9831	0.9881	0.9985	0.9988	0.9982	0.9997
Ejecución 6	0.371	0.4642	0.9838	0.9737	0.9985	0.9844	0.9977	0.9996
Ejecución 7	0.3571	0.4655	0.9644	0.9743	0.9996	0.9992	0.997	0.9984
Ejecución 8	0.3678	0.4518	0.9803	0.9661	0.9997	0.9992	0.9996	0.9996
Ejecución 9	0.3648	0.4396	0.9767	0.9823	0.9996	0.9861	0.9988	0.9992
Ejecución 10	0.3668	0.455	0.9729	0.9718	0.9982	0.9996	0.9992	0.9992
Segment	0.2303 \pm 0.0319	0.2606 \pm 0.0102	0.1436 \pm 0.0558	0.1808 \pm 0.0565	0.6213 \pm 0.1895	0.7095 \pm 0.1906	0.9452 \pm 0.0503	0.9897 \pm 0.0146
Ejecución 1	0.2634	0.2616	0.1822	0.228	0.8259	0.8841	0.8631	0.9932
Ejecución 2	0.1962	0.2448	0.1174	0.2173	0.3433	0.6507	0.9933	0.9966
Ejecución 3	0.1968	0.2745	0.1488	0.0617	0.8944	0.7256	0.9831	0.9932
Ejecución 4	0.2107	0.2708	0.1275	0.1816	0.7314	0.9139	0.8811	0.9474
Ejecución 5	0.1896	0.2712	0.2107	0.1867	0.7201	0.6799	0.9829	1.0
Ejecución 6	0.2636	0.2522	0.0847	0.204	0.58	0.8928	0.9865	0.9966
Ejecución 7	0.2627	0.2654	0.1163	0.1505	0.2854	0.7379	0.8865	0.9864
Ejecución 8	0.2021	0.2619	0.0799	0.251	0.5849	0.7172	0.983	0.9933
Ejecución 9	0.2675	0.2436	0.1048	0.1038	0.506	0.685	0.9832	1.0
Ejecución 10	0.2502	0.26	0.2638	0.2234	0.7416	0.2081	0.9092	0.9899
Sonar	0.0002 \pm 0.0068	-0.0045 \pm 0.0003	0.071 \pm 0.083	0.0734 \pm 0.0983	0.492 \pm 0.2155	0.478 \pm 0.2416	0.7801 \pm 0.3927	0.6542 \pm 0.4349
Ejecución 1	0.0065	-0.0039	0.1669	-0.0003	0.4768	0.3877	1.0	0.9808
Ejecución 2	-0.0046	-0.0039	-0.0044	0.0401	0.7815	0.7477	1.0	1.0
Ejecución 3	-0.0024	-0.0046	-0.0033	0.0011	0.2274	0.3186	1.0	1.0
Ejecución 4	-0.0039	-0.0047	-0.0048	-0.0047	0.3877	0.6198	0.003	1.0
Ejecución 5	-0.0032	-0.0046	0.167	0.0362	0.1751	0.7815	1.0	0.0014
Ejecución 6	0.0162	-0.0044	0.1833	0.0027	0.3999	0.2093	1.0	0.0363
Ejecución 7	-0.0047	-0.0047	-0.0015	0.2004	0.3999	0.731	0.9808	1.0
Ejecución 8	-0.0044	-0.0046	-0.0044	0.297	0.8336	0.2762	0.0014	1.0
Ejecución 9	0.0065	-0.0044	0.0443	0.0253	0.4902	0.0734	0.8161	0.4504
Ejecución 10	-0.0043	-0.0047	0.167	0.1365	0.7477	0.6352	1.0	0.0735
Soybean	0.5862 \pm 0.0146	0.5484 \pm 0.0442	0.6002 \pm 0.138	0.5805 \pm 0.0946	0.6452 \pm 0.1736	0.6629 \pm 0.1398	0.4934 \pm 0.0658	0.46 \pm 0.0764
Ejecución 1	0.5519	0.4768	0.6097	0.529	0.7047	0.5401	0.5513	0.5228
Ejecución 2	0.5855	0.4917	0.5611	0.6077	0.4613	0.5733	0.4699	0.4977
Ejecución 3	0.5855	0.5678	0.5611	0.5741	0.5863	0.5401	0.5694	0.3026
Ejecución 4	0.5855	0.5814	0.5611	0.4765	1.0	0.9533	0.4771	0.4836
Ejecución 5	0.5855	0.5634	0.3076	0.4807	0.5508	0.5216	0.3831	0.4752
Ejecución 6	0.5855	0.5814	0.82	0.6527	0.6984	0.6242	0.5261	0.464
Ejecución 7	0.5855	0.4768	0.6097	0.539	0.6208	0.8758	0.4642	0.3335
Ejecución 8	0.5855	0.5814	0.82	0.8201	0.8795	0.6859	0.3881	0.5162
Ejecución 9	0.6153	0.5814	0.6097	0.5485	0.5446	0.6049	0.5272	0.551
Ejecución 10	0.5963	0.5814	0.5423	0.5762	0.4058	0.7095	0.5775	0.4529
Spambase	0.0606 \pm 0.0035	0.0732 \pm 0.0035	0.8952 \pm 0.2978	0.9945 \pm 0.0009	1.0 \pm 0.0	1.0 \pm 0.0	1.0 \pm 0.0	1.0 \pm 0.0
Ejecución 1	0.058	0.0746	0.0018	0.9942	1.0	1.0	1.0	1.0
Ejecución 2	0.0624	0.0697	0.9942	0.9942	1.0	1.0	1.0	1.0
Ejecución 3	0.0646	0.0713	0.9942	0.9942	1.0	1.0	1.0	1.0
Ejecución 4	0.059	0.0668	0.9942	0.9942	1.0	1.0	1.0	1.0
Ejecución 5	0.0573	0.0731	0.9942	0.9942	1.0	1.0	1.0	1.0
Ejecución 6	0.0572	0.0734	0.9942	0.9971	1.0	1.0	1.0	1.0
Ejecución 7	0.0661	0.0766	0.9942	0.9942	1.0	1.0	1.0	1.0
Ejecución 8	0.0572	0.0774	0.9942	0.9942	1.0	1.0	1.0	1.0

Continúa en la siguiente página

A. Resultados experimentales completos.

Tabla A.1 – continúa de la página anterior

Nombre	0 % restricciones		10 % restricciones		15 % restricciones		20 % restricciones	
	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D
Ejecución 9	0.0656	0.071	0.9942	0.9942	1.0	1.0	1.0	1.0
Ejecución 10	0.0583	0.0783	0.9971	0.9942	1.0	1.0	1.0	1.0
Spectheart	-0.0997 ± 0.0022	-0.105 ± 0.001	0.4049 ± 0.1527	0.31 ± 0.1565	0.9558 ± 0.082	0.8944 ± 0.266	0.9762 ± 0.0397	0.9073 ± 0.2615
Ejecución 1	-0.0996	-0.1048	0.3942	0.4007	0.9831	0.9831	1.0	1.0
Ejecución 2	-0.0996	-0.1066	0.4781	0.4582	0.9831	0.9831	1.0	1.0
Ejecución 3	-0.0984	-0.1034	0.4784	0.309	0.7098	0.9831	1.0	1.0
Ejecución 4	-0.1017	-0.1041	0.3571	0.5434	0.9831	0.9831	1.0	1.0
Ejecución 5	-0.0984	-0.1053	0.5756	0.0114	0.9831	0.9831	1.0	1.0
Ejecución 6	-0.0957	-0.1041	0.5644	0.0792	0.9831	0.9831	1.0	1.0
Ejecución 7	-0.1041	-0.1053	0.4507	0.3167	0.9831	0.9831	0.9319	1.0
Ejecución 8	-0.0984	-0.1048	0.0701	0.4335	0.9831	0.9831	0.8808	1.0
Ejecución 9	-0.1017	-0.1067	0.1938	0.2702	0.9831	0.9831	0.9489	0.9489
Ejecución 10	-0.0996	-0.1053	0.4862	0.278	0.9831	0.9831	1.0	0.1243
Tae	0.0057 ± 0.0041	0.0203 ± 0.0172	0.0286 ± 0.0195	0.014 ± 0.023	0.0771 ± 0.0415	0.0614 ± 0.0409	0.5868 ± 0.121	0.5706 ± 0.1245
Ejecución 1	0.0018	0.0372	0.04	0.0054	0.1368	0.0291	0.6487	0.8278
Ejecución 2	0.0068	0.0399	0.0072	0.0273	0.0192	0.0528	0.6057	0.6057
Ejecución 3	0.0069	-0.0046	0.0429	-0.0036	0.0777	0.1439	0.4253	0.4426
Ejecución 4	0.0052	-0.0052	0.038	0.024	0.0653	0.0288	0.4933	0.7557
Ejecución 5	0.0089	0.0369	0.0186	0.0711	0.0314	0.1176	0.7717	0.4882
Ejecución 6	0.0027	0.0327	0.044	-0.0098	0.1391	0.0763	0.652	0.5454
Ejecución 7	0.0086	-0.0019	0.0632	0.0114	0.1101	0.0481	0.537	0.4139
Ejecución 8	0.0144	0.03	0.0089	0.0079	0.0239	0.0445	0.5619	0.5089
Ejecución 9	0.0017	0.019	0.0265	0.0034	0.0949	0.073	0.7336	0.5366
Ejecución 10	-0.0001	0.019	-0.0033	-0.0057	0.0726	-0.0002	0.3875	0.5809
Thyroid	-0.0251 ± 0.0118	-0.0077 ± 0.001	0.9901 ± 0.0009	0.9896 ± 0.001	0.995 ± 0.0023	0.9969 ± 0.0013	0.9431 ± 0.1594	0.9992 ± 0.0004
Ejecución 1	-0.0121	-0.007	0.9896	0.9895	0.9936	0.9942	0.9946	0.9989
Ejecución 2	-0.0329	-0.0096	0.9897	0.9895	0.9891	0.9973	0.9994	0.9994
Ejecución 3	-0.0328	-0.0084	0.9895	0.9886	0.9961	0.9966	0.9994	0.9994
Ejecución 4	-0.0321	-0.007	0.9892	0.991	0.9944	0.9983	0.9924	0.9997
Ejecución 5	-0.0321	-0.0084	0.9904	0.989	0.9957	0.9976	1.0	0.9992
Ejecución 6	-0.0328	-0.0078	0.9897	0.9888	0.995	0.9971	0.9978	0.9992
Ejecución 7	-0.0012	-0.0072	0.9908	0.9894	0.9976	0.9981	0.4651	0.9984
Ejecución 8	-0.0339	-0.0086	0.989	0.9886	0.9946	0.9976	0.9994	0.9994
Ejecución 9	-0.0321	-0.0067	0.9917	0.9895	0.9966	0.995	0.9908	0.9994
Ejecución 10	-0.0091	-0.006	0.9914	0.9919	0.9971	0.9973	0.9917	0.9989
Titanic	0.0512 ± 0.0814	0.1484 ± 0.0484	0.9795 ± 0.041	0.9926 ± 0.0024	0.9914 ± 0.0049	0.9914 ± 0.0073	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	-0.0035	0.1095	0.9939	0.9939	0.9939	0.9939	1.0	1.0
Ejecución 2	0.2127	0.2159	0.8568	0.9939	0.9939	0.9939	1.0	1.0
Ejecución 3	0.1856	0.2125	0.9939	0.9939	0.9939	0.9694	1.0	1.0
Ejecución 4	-0.0035	0.1095	0.9939	0.9877	0.9816	0.9939	1.0	1.0
Ejecución 5	-0.0035	0.1095	0.9939	0.9939	0.9816	0.9939	1.0	1.0
Ejecución 6	0.0248	0.2159	0.9877	0.9939	0.9939	0.9939	1.0	1.0
Ejecución 7	-0.0035	0.1095	0.9939	0.9877	0.9939	0.9939	1.0	1.0
Ejecución 8	0.1095	0.1095	0.9939	0.9939	0.9939	0.9939	1.0	1.0
Ejecución 9	-0.0035	0.1822	0.9939	0.9939	0.9939	0.9939	1.0	1.0
Ejecución 10	-0.0035	0.1095	0.9939	0.9939	0.9939	0.9939	1.0	1.0
Vehicle	0.0894 ± 0.0053	0.0695 ± 0.0036	0.4089 ± 0.0927	0.4424 ± 0.1433	0.9513 ± 0.1022	0.9857 ± 0.0028	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	0.08	0.0699	0.3636	0.3139	0.9845	0.9814	1.0	1.0
Ejecución 2	0.0882	0.0771	0.4291	0.5909	0.9814	0.9814	1.0	1.0
Ejecución 3	0.0888	0.0713	0.199	0.376	0.9813	0.9876	1.0	1.0
Ejecución 4	0.088	0.0714	0.4552	0.5158	0.9814	0.9876	1.0	1.0
Ejecución 5	0.0925	0.0708	0.4276	0.3567	0.9871	0.9904	1.0	1.0
Ejecución 6	0.0803	0.0701	0.3134	0.4087	0.9904	0.9845	1.0	1.0
Ejecución 7	0.0937	0.065	0.417	0.1699	0.9904	0.9876	1.0	1.0
Ejecución 8	0.0952	0.0648	0.5466	0.6822	0.9904	0.9876	1.0	1.0
Ejecución 9	0.091	0.0693	0.4406	0.5714	0.9814	0.9842	1.0	1.0
Ejecución 10	0.096	0.065	0.4975	0.4382	0.645	0.9845	1.0	1.0
Wdbc	0.2824 ± 0.0212	0.3576 ± 0.0122	0.9782 ± 0.011	0.9831 ± 0.0109	1.0 ± 0.0	0.9958 ± 0.0035	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	0.2537	0.3631	0.9719	0.9929	1.0	1.0	1.0	1.0
Ejecución 2	0.3089	0.3538	0.9789	0.9789	1.0	0.9929	1.0	1.0
Ejecución 3	0.2872	0.3447	0.9719	0.9789	1.0	0.9929	1.0	1.0
Ejecución 4	0.2787	0.3724	0.9929	0.9929	1.0	1.0	1.0	1.0
Ejecución 5	0.2456	0.3447	0.9581	0.9859	1.0	0.9929	1.0	1.0
Ejecución 6	0.3045	0.3771	0.965	0.9719	1.0	0.9929	1.0	1.0
Ejecución 7	0.2703	0.3401	0.9929	0.9719	1.0	0.9929	1.0	1.0
Ejecución 8	0.3045	0.3492	0.9789	0.9929	1.0	0.9929	1.0	1.0
Ejecución 9	0.3002	0.3681	0.9859	0.965	1.0	1.0	1.0	1.0
Ejecución 10	0.2703	0.3631	0.9859	1.0	1.0	1.0	1.0	1.0
Wine	0.2147 ± 0.0112	0.332 ± 0.0134	0.3277 ± 0.0987	0.3487 ± 0.0988	0.5114 ± 0.0998	0.4558 ± 0.0951	0.8571 ± 0.1165	0.9013 ± 0.0701
Ejecución 1	0.2171	0.3036	0.4282	0.5063	0.5391	0.5046	0.6742	0.9501
Ejecución 2	0.2294	0.3512	0.418	0.1726	0.383	0.3635	0.9651	0.9503
Ejecución 3	0.2156	0.3334	0.2679	0.3238	0.6138	0.299	0.7966	0.8103
Ejecución 4	0.2085	0.3395	0.2294	0.39	0.509	0.6126	0.8933	0.9503
Ejecución 5	0.2047	0.339	0.4298	0.5039	0.6351	0.4665	0.7263	0.9651
Ejecución 6	0.2261	0.3263	0.2548	0.3944	0.6163	0.552	0.9468	0.9356
Ejecución 7	0.2206	0.3434	0.1243	0.27	0.5378	0.4653	0.95	0.9121
Ejecución 8	0.193	0.3416	0.3485	0.2945	0.4279	0.387	0.9651	0.7552
Ejecución 9	0.2047	0.3193	0.3632	0.2845	0.3183	0.533	0.9651	0.8339
Ejecución 10	0.2272	0.3227	0.4124	0.3465	0.5339	0.3544	0.6881	0.9503
Zoo	0.6616 ± 0.0267	0.6368 ± 0.018	0.4813 ± 0.0849	0.439 ± 0.0791	0.4699 ± 0.1003	0.5234 ± 0.0936	0.6423 ± 0.1099	0.5335 ± 0.1389
Ejecución 1	0.6654	0.6631	0.3564	0.5168	0.3965	0.5175	0.5671	0.4304
Ejecución 2	0.6154	0.6331	0.5426	0.5885	0.3801	0.586	0.7594	0.6227
Ejecución 3	0.6875	0.638	0.4304	0.4792	0.6765	0.4433	0.6885	0.7574
Ejecución 4	0.6719	0.626	0.5266	0.2915	0.5154	0.5684	0.5647	0.3814
Ejecución 5	0.6596	0.6545	0.534	0.4017	0.4964	0.4717	0.5156	0.389
Ejecución 6	0.6681	0.6056	0.5632	0.3639	0.3413	0.4619	0.7335	0.6771
Ejecución 7	0.6488	0.615	0.3343	0.4279	0.4193	0.3911	0.4418	0.5485
Ejecución 8	0.6157	0.6377	0.6052	0.4338	0.4862	0.7024	0.7852	0.3674
Ejecución 9	0.697	0.6621	0.4413	0.4854	0.5967	0.4502	0.733	0.6978
Ejecución 10	0.6863	0.6331	0.479	0.4012	0.3909	0.6412	0.6342	0.4636
Media	0.1708 ± 0.2111	0.1903 ± 0.225	0.5558 ± 0.3506	0.5829 ± 0.3436	0.7725 ± 0.2724	0.8044 ± 0.2672	0.8926 ± 0.1446	0.8942 ± 0.1583

Tabla A.2.: Resultados de las ejecuciones del CS2S.

Nombre	0% restricciones		10% restricciones		15% restricciones		20% restricciones	
	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D
Appendicitis	0.5306 ± 0.0153	0.5041 ± 0.0236	0.456 ± 0.0304	0.4142 ± 0.0428	0.4405 ± 0.037	0.4513 ± 0.0184	0.4147 ± 0.0448	0.4839 ± 0.026
Ejecución 1	0.5458	0.5255	0.4108	0.3724	0.4618	0.4618	0.4861	0.5255
Ejecución 2	0.5153	0.4688	0.4582	0.4688	0.4618	0.4194	0.4194	0.4618
Ejecución 3	0.5153	0.5255	0.4965	0.3724	0.4618	0.4618	0.3765	0.4861
Ejecución 4	0.5458	0.4965	0.4582	0.443	0.3765	0.4618	0.3765	0.4618
Ejecución 5	0.5481	0.5415	0.4645	0.4815	0.4345	0.515	0.4201	0.4703
Ejecución 6	0.5302	0.5134	0.4146	0.4547	0.4784	0.4301	0.4102	0.489
Ejecución 7	0.5907	0.4985	0.4954	0.3124	0.4504	0.4054	0.435	0.4025
Ejecución 8	0.4972	0.4712	0.3978	0.3033	0.415	0.554	0.415	0.4385
Ejecución 9	0.4781	0.5348	0.5154	0.3524	0.4024	0.497	0.3804	0.4815
Ejecución 10	0.5712	0.5645	0.5097	0.4106	0.4845	0.4302	0.3901	0.4908
Balance	0.0736 ± 0.0149	0.0571 ± 0.041	0.1443 ± 0.0508	0.2914 ± 0.0616	0.2711 ± 0.1123	0.3253 ± 0.1061	0.1122 ± 0.0298	0.1539 ± 0.0435
Ejecución 1	0.064	0.0565	0.1526	0.2467	0.1108	0.1759	0.1482	0.1611
Ejecución 2	0.0877	0.0249	0.1915	0.3347	0.3003	0.429	0.0653	0.0946
Ejecución 3	0.0543	0.0225	0.0596	0.2168	0.4242	0.4218	0.119	0.1434
Ejecución 4	0.0884	0.124	0.1735	0.3672	0.2489	0.2743	0.1159	0.2164
Ejecución 5	0.0712	0.0571	0.1678	0.3214	0.2978	0.3215	0.1271	0.167
Ejecución 6	0.0691	0.0671	0.1054	0.2198	0.3784	0.2945	0.1264	0.1507
Ejecución 7	0.0783	0.0487	0.0645	0.3173	0.2145	0.3027	0.138	0.1438
Ejecución 8	0.0749	0.0541	0.1514	0.2354	0.2517	0.3878	0.1002	0.1207
Ejecución 9	0.0755	0.0498	0.1807	0.2649	0.2847	0.2014	0.1015	0.1708
Ejecución 10	0.0765	0.0671	0.1348	0.2506	0.3345	0.3498	0.1021	0.156
Banana	0.0009 ± 0.0013	0.0017 ± 0.0043	0.1194 ± 0.0149	0.1815 ± 0.072	0.0885 ± 0.0367	0.1751 ± 0.0342	0.0947 ± 0.0245	0.175 ± 0.0184
Ejecución 1	0.0016	0.0	0.1434	0.0691	0.0989	0.1801	0.0959	0.1634
Ejecución 2	0.0027	-0.0013	0.1059	0.1728	0.1183	0.1945	0.1002	0.201
Ejecución 3	-0.0004	-0.0009	0.1078	0.2607	0.1105	0.1182	0.1254	0.1825
Ejecución 4	-0.0002	0.009	0.1202	0.2233	0.026	0.2074	0.0569	0.1528
Ejecución 5	-0.0001	0.0014	0.1187	0.1214	0.0789	0.1802	0.084	0.1875
Ejecución 6	-0.0002	0.0021	0.2014	0.2648	0.0815	0.167	0.1055	0.164
Ejecución 7	-0.0003	-0.0015	0.2012	0.2514	0.0981	0.1309	0.1021	0.1754
Ejecución 8	-0.0001	0.0021	0.1254	0.1454	0.0871	0.1819	0.1102	0.1845
Ejecución 9	0.0024	-0.0014	0.1054	0.1564	0.0812	0.2108	0.0901	0.154
Ejecución 10	0.0019	0.0006	0.1607	0.2107	0.0831	0.1309	0.0948	0.1648
Breast_cancer	0.3038 ± 0.0246	0.3614 ± 0.0356	0.6369 ± 0.0146	0.6377 ± 0.0255	0.6369 ± 0.022	0.607 ± 0.011	0.6399 ± 0.019	0.6485 ± 0.0232
Ejecución 1	0.2958	0.331	0.6348	0.6323	0.6531	0.6118	0.6524	0.6384
Ejecución 2	0.2958	0.4057	0.6364	0.6758	0.6478	0.6175	0.6254	0.6272
Ejecución 3	0.2787	0.3865	0.6585	0.6044	0.5989	0.5885	0.6175	0.6407
Ejecución 4	0.3446	0.3221	0.6175	0.638	0.6478	0.61	0.664	0.6876
Ejecución 5	0.2904	0.3456	0.5954	0.6324	0.6154	0.6048	0.6028	0.7504
Ejecución 6	0.2804	0.2974	0.5104	0.6158	0.5915	0.6124	0.6048	0.6904
Ejecución 7	0.3105	0.2845	0.6245	0.6174	0.5845	0.592	0.5209	0.6708
Ejecución 8	0.3708	0.3154	0.6564	0.6627	0.6245	0.6012	0.509	0.6505
Ejecución 9	0.2105	0.3948	0.6045	0.5692	0.6587	0.6378	0.6107	0.645
Ejecución 10	0.3001	0.3648	0.6401	0.61	0.6403	0.6001	0.6605	0.6301
Contraceptive	0.0107 ± 0.0017	0.0232 ± 0.0026	0.0185 ± 0.0099	0.0251 ± 0.0008	0.0195 ± 0.0079	0.0244 ± 0.0044	0.0245 ± 0.0035	0.0266 ± 0.001
Ejecución 1	0.012	0.027	0.0219	0.0244	0.0174	0.0239	0.0194	0.0257
Ejecución 2	0.0108	0.0198	0.0187	0.0253	0.0075	0.0186	0.0267	0.0258
Ejecución 3	0.0078	0.022	0.0304	0.0243	0.0252	0.0309	0.0286	0.0282
Ejecución 4	0.012	0.0238	0.003	0.0263	0.0278	0.0241	0.0232	0.0264
Ejecución 5	0.0097	0.0264	0.0015	0.0214	0.0234	0.0214	0.0248	0.0204
Ejecución 6	0.0087	0.0215	0.0034	0.0755	0.0187	0.031	0.0235	0.0315
Ejecución 7	0.0215	0.0197	0.0012	0.0365	0.0214	0.0195	0.0254	0.0201
Ejecución 8	0.0105	0.0275	0.0009	0.0167	0.0244	0.0215	0.0275	0.0198
Ejecución 9	0.01	0.0225	0.0015	0.146	0.0197	0.0254	0.0278	0.0287
Ejecución 10	0.0094	0.0198	0.0013	0.0697	0.0234	0.0264	0.0198	0.032
Ecoli	0.2275 ± 0.0262	0.2784 ± 0.0407	0.4107 ± 0.0198	0.3189 ± 0.0255	0.4558 ± 0.0142	0.4066 ± 0.0169	0.4103 ± 0.0546	0.3522 ± 0.0445
Ejecución 1	0.1901	0.2856	0.4042	0.3219	0.4565	0.4089	0.4636	0.3221
Ejecución 2	0.2641	0.327	0.4327	0.291	0.4513	0.4243	0.4289	0.4289
Ejecución 3	0.2268	0.214	0.4242	0.3587	0.4379	0.414	0.4298	0.3327
Ejecución 4	0.2287	0.2868	0.3815	0.3039	0.4775	0.3789	0.3189	0.325
Ejecución 5	0.2194	0.2645	0.4158	0.3632	0.4687	0.4054	0.441	0.3064
Ejecución 6	0.2168	0.2456	0.3678	0.2761	0.4345	0.3904	0.4039	0.318
Ejecución 7	0.2504	0.2871	0.3972	0.3214	0.4054	0.3851	0.3901	0.397
Ejecución 8	0.3015	0.2015	0.4215	0.2954	0.4982	0.4105	0.3804	0.284
Ejecución 9	0.1908	0.3048	0.3884	0.326	0.3954	0.4112	0.4105	0.345
Ejecución 10	0.2315	0.3102	0.3809	0.3025	0.4751	0.4275	0.432	0.3804
Glass	0.2255 ± 0.0066	0.2268 ± 0.0148	0.2345 ± 0.0078	0.2379 ± 0.0133	0.245 ± 0.0123	0.2526 ± 0.0186	0.2438 ± 0.0206	0.2675 ± 0.0242
Ejecución 1	0.2202	0.2135	0.2354	0.2521	0.2443	0.2657	0.2084	0.2574
Ejecución 2	0.2242	0.2518	0.2375	0.2334	0.2471	0.2758	0.2581	0.2357
Ejecución 3	0.2366	0.2219	0.2431	0.2181	0.2615	0.2312	0.2579	0.3018
Ejecución 4	0.2208	0.2197	0.2219	0.2478	0.227	0.2376	0.2505	0.2748
Ejecución 5	0.2249	0.2287	0.2404	0.2114	0.2124	0.264	0.2504	0.2546
Ejecución 6	0.2235	0.2017	0.2545	0.1486	0.2015	0.2348	0.2365	0.2405
Ejecución 7	0.2015	0.197	0.3045	0.2528	0.2984	0.3105	0.2642	0.2548
Ejecución 8	0.1908	0.2485	0.2987	0.3054	0.3105	0.278	0.2504	0.2481
Ejecución 9	0.3017	0.2378	0.2045	0.1934	0.3312	0.2691	0.2648	0.254
Ejecución 10	0.172	0.2141	0.2784	0.2706	0.215	0.2301	0.2154	0.2647
Haberman	-0.0212 ± 0.0161	-0.0225 ± 0.0097	0.0207 ± 0.0107	0.2148 ± 0.0131	0.1487 ± 0.0299	0.187 ± 0.0533	0.1889 ± 0.0221	0.1702 ± 0.0441
Ejecución 1	-0.009	-0.0255	0.207	0.237	0.1104	0.2181	0.1875	0.1208
Ejecución 2	-0.0142	-0.0265	0.2004	0.2107	0.1474	0.2162	0.2091	0.2189
Ejecución 3	-0.0489	-0.0061	0.1868	0.2043	0.1425	0.2189	0.2056	0.2091
Ejecución 4	-0.0125	-0.0316	0.2162	0.207	0.1943	0.0946	0.1533	0.1318
Ejecución 5	-0.021	-0.0274	0.2124	0.2604	0.1564	0.197	0.1907	0.1807
Ejecución 6	-0.0209	-0.0197	0.1978	0.301	0.1054	0.172	0.1701	0.1654
Ejecución 7	-0.0348	-0.0238	0.2084	0.2126	0.1987	0.097	0.1358	0.1405

Continúa en la siguiente página

A. Resultados experimentales completos.

Tabla A.2 – continúa de la página anterior								
Nombre	0 % restricciones		10 % restricciones		15 % restricciones		20 % restricciones	
	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D
Ejecución 8	-0.051	-0.0315	0.2315	0.1847	0.1054	0.2401	0.2011	0.2018
Ejecución 9	-0.007	-0.0187	0.1975	0.2448	0.0974	0.1705	0.2105	0.1307
Ejecución 10	-0.0067	-0.0215	0.189	0.2044	0.1845	0.1894	0.1547	0.1928
Hayes_roth	0.0776 ± 0.0152	0.0364 ± 0.0348	0.0215 ± 0.0144	0.0345 ± 0.0172	0.0669 ± 0.0396	0.0818 ± 0.0247	0.089 ± 0.0637	0.1047 ± 0.0347
Ejecución 1	0.0938	-0.0194	0.0385	0.0611	0.0573	0.0446	0.1883	0.0575
Ejecución 2	0.0802	0.0708	-0.0013	0.0323	0.09	0.0833	0.0579	0.1075
Ejecución 3	0.0526	0.0346	0.0245	0.0315	0.0074	0.0851	0.0942	0.155
Ejecución 4	0.0837	0.0596	0.0244	0.0129	0.1128	0.1142	0.0156	0.0985
Ejecución 5	0.045	0.0294	0.0215	0.0379	0.078	0.091	0.098	0.0978
Ejecución 6	0.0848	0.0305	0.0198	0.0326	0.0978	0.1101	0.089	0.087
Ejecución 7	0.0902	0.0294	0.0245	0.0292	0.0643	0.1024	0.0981	0.0908
Ejecución 8	0.0846	0.0378	0.0102	0.0279	0.0345	0.064	0.1058	0.0804
Ejecución 9	0.0531	0.0394	0.0198	0.0327	0.0289	0.0504	0.101	0.1024
Ejecución 10	0.0643	0.0345	0.0245	0.0354	0.0756	0.0823	0.098	0.135
Heart	0.0022 ± 0.0013	0.0071 ± 0.0028	0.0448 ± 0.0261	0.041 ± 0.0277	0.0978 ± 0.0257	0.0935 ± 0.015	0.0837 ± 0.0178	0.0813 ± 0.0077
Ejecución 1	0.0024	0.0073	0.0458	0.007	0.1075	0.082	0.0624	0.0716
Ejecución 2	0.004	0.0024	0.0676	0.0363	0.0975	0.0777	0.0714	0.0887
Ejecución 3	0.0014	0.0094	0.0018	0.0844	0.1285	0.1159	0.0934	0.0888
Ejecución 4	0.0007	0.0091	0.0636	0.0362	0.0576	0.0981	0.1076	0.0757
Ejecución 5	0.0018	0.0074	0.0454	0.0494	0.0978	0.089	0.0708	0.089
Ejecución 6	0.0022	0.0097	0.0489	0.0232	0.0878	0.094	0.0804	0.098
Ejecución 7	0.0024	0.0064	0.0413	0.0316	0.0647	0.0934	0.098	0.0742
Ejecución 8	0.0023	0.0054	0.0345	0.0567	0.1024	0.079	0.0718	0.0564
Ejecución 9	0.0018	0.0109	0.0645	0.0165	0.0971	0.0942	0.0848	0.0804
Ejecución 10	0.0019	0.0063	0.0544	0.064	0.087	0.0994	0.0798	0.0705
Ionosphere	-0.0153 ± 0.0126	0.0245 ± 0.0246	0.2746 ± 0.1119	0.2476 ± 0.0742	0.2933 ± 0.0812	0.289 ± 0.0871	0.3571 ± 0.0402	0.241 ± 0.0303
Ejecución 1	-0.0074	-0.0141	0.0852	0.2845	0.3406	0.1626	0.3534	0.2014
Ejecución 2	-0.037	0.0343	0.3049	0.1255	0.2597	0.2774	0.2969	0.2646
Ejecución 3	-0.0091	0.0247	0.3727	0.2573	0.1793	0.3088	0.4091	0.2755
Ejecución 4	-0.0074	0.0533	0.3353	0.323	0.3934	0.4068	0.3687	0.2223
Ejecución 5	0.0159	0.0234	0.3454	0.2617	0.3214	0.3015	0.3045	0.2506
Ejecución 6	0.0137	0.0197	0.2508	0.1486	0.2901	0.3184	0.3405	0.2608
Ejecución 7	0.0183	0.0217	0.2954	0.1269	0.2782	0.2994	0.3849	0.243
Ejecución 8	0.0207	0.0187	0.3754	0.2354	0.3245	0.2978	0.3974	0.2358
Ejecución 9	0.0092	0.0288	0.3048	0.2903	0.3451	0.3405	0.354	0.2335
Ejecución 10	0.0169	0.0209	0.2946	0.1128	0.2978	0.2751	0.3405	0.2754
Iris	0.6166 ± 0.067	0.695 ± 0.032	0.6691 ± 0.0208	0.7116 ± 0.0598	0.7424 ± 0.0946	0.7576 ± 0.1071	0.7082 ± 0.0067	0.7555 ± 0.0765
Ejecución 1	0.6318	0.6521	0.6615	0.7711	0.6615	0.5752	0.7015	0.6066
Ejecución 2	0.55	0.7322	0.688	0.641	0.9037	0.8508	0.7149	0.7155
Ejecución 3	0.5647	0.677	0.6383	0.7707	0.7021	0.8016	0.7015	0.7302
Ejecución 4	0.7195	0.7183	0.6885	0.6637	0.7021	0.8026	0.7149	0.8856
Ejecución 5	0.559	0.7015	0.6544	0.6321	0.7154	0.841	0.7081	0.754
Ejecución 6	0.5904	0.697	0.6874	0.7615	0.7054	0.7604	0.6904	0.8094
Ejecución 7	0.5801	0.681	0.6389	0.6514	0.6954	0.7301	0.7201	0.7308
Ejecución 8	0.6258	0.6912	0.6887	0.6707	0.6851	0.7915	0.7251	0.9304
Ejecución 9	0.5803	0.7215	0.6987	0.692	0.7545	0.7051	0.6801	0.6048
Ejecución 10	0.671	0.7504	0.6478	0.741	0.7851	0.6915	0.7214	0.7507
Monk2	0.0687 ± 0.0498	0.1242 ± 0.0349	0.262 ± 0.0614	0.2794 ± 0.0835	0.2939 ± 0.0886	0.2221 ± 0.0594	0.3122 ± 0.0371	0.2681 ± 0.0664
Ejecución 1	0.0318	0.0718	0.3659	0.3391	0.4064	0.1422	0.3591	0.1837
Ejecución 2	0.1544	0.1515	0.2171	0.3747	0.3036	0.2606	0.3029	0.3045
Ejecución 3	0.0429	0.1598	0.2164	0.162	0.158	0.1908	0.3286	0.3554
Ejecución 4	0.0454	0.1134	0.2484	0.2417	0.3074	0.2948	0.2579	0.2283
Ejecución 5	0.0672	0.1535	0.2978	0.2632	0.315	0.2143	0.3205	0.2705
Ejecución 6	0.0723	0.0978	0.2184	0.3464	0.2815	0.2405	0.3405	0.2215
Ejecución 7	0.0703	0.0997	0.2456	0.1725	0.3154	0.2871	0.3105	0.2804
Ejecución 8	0.0696	0.1632	0.2354	0.2474	0.2895	0.2015	0.2908	0.3043
Ejecución 9	0.0604	0.1145	0.2784	0.2799	0.2964	0.291	0.278	0.31
Ejecución 10	0.0618	0.0978	0.268	0.2476	0.3217	0.1972	0.2908	0.245
Newthyroid	0.4271 ± 0.0834	0.3569 ± 0.1483	0.4329 ± 0.0915	0.3621 ± 0.0197	0.5516 ± 0.0709	0.5183 ± 0.0307	0.4817 ± 0.0984	0.4004 ± 0.0805
Ejecución 1	0.5208	0.508	0.5639	0.3421	0.43	0.5346	0.453	0.3808
Ejecución 2	0.4447	0.3507	0.3278	0.389	0.5792	0.5535	0.4245	0.3787
Ejecución 3	0.2922	0.4495	0.4694	0.3725	0.6079	0.5138	0.649	0.531
Ejecución 4	0.4505	0.1192	0.37	0.3444	0.589	0.471	0.4	0.311
Ejecución 5	0.4212	0.4176	0.4354	0.3497	0.5407	0.6018	0.4907	0.4055
Ejecución 6	0.4505	0.3945	0.4156	0.3197	0.5879	0.5915	0.5048	0.4302
Ejecución 7	0.2082	0.3457	0.3945	0.2954	0.5045	0.517	0.5107	0.3907
Ejecución 8	0.3913	0.3215	0.5145	0.3702	0.5987	0.5381	0.5408	0.388
Ejecución 9	0.5194	0.378	0.4024	0.3109	0.6015	0.4975	0.4608	0.3951
Ejecución 10	0.4323	0.2987	0.3485	0.3079	0.6205	0.4124	0.4701	0.4201
Page_blocks	-0.0202 ± 0.0066	-0.0208 ± 0.0085	0.0626 ± 0.0128	0.0459 ± 0.0458	0.0389 ± 0.023	0.0486 ± 0.0472	0.0902 ± 0.0341	0.0221 ± 0.0168
Ejecución 1	-0.0145	-0.0303	0.0663	-0.0001	-0.0007	-0.002	0.0859	0.0212
Ejecución 2	-0.0159	-0.0247	0.0409	0.0288	0.0493	0.1209	0.0732	0.0098
Ejecución 3	-0.0188	-0.0072	0.0694	0.1221	0.0506	0.0582	0.0554	0.0497
Ejecución 4	-0.0313	-0.0209	0.0736	0.0328	0.0563	0.018	0.1461	0.0074
Ejecución 5	-0.0186	-0.0197	0.0504	0.0147	0.045	0.045	0.098	0.0234
Ejecución 6	-0.0195	-0.0245	0.0498	0.0756	0.0345	0.0439	0.1088	0.024
Ejecución 7	-0.0248	-0.0197	0.0678	0.0517	0.047	0.0328	0.1021	0.0264
Ejecución 8	-0.021	-0.0193	0.054	0.0443	0.0124	0.0394	0.0897	0.0197
Ejecución 9	-0.022	-0.0248	0.0354	0.0621	0.045	0.0416	0.098	0.0187
Ejecución 10	-0.0183	-0.0218	0.0898	0.0331	0.0291	0.0512	0.087	0.0235
Phoneme	0.0744 ± 0.1434	0.2774 ± 0.0052	0.2878 ± 0.0116	0.2568 ± 0.0552	0.2854 ± 0.0027	0.2379 ± 0.0514	0.2599 ± 0.0432	0.2502 ± 0.0443
Ejecución 1	-0.0734	0.2806	0.2906	0.2888	0.2896	0.2621	0.1872	0.2694
Ejecución 2	0.1966	0.2838	0.2723	0.3041	0.2859	0.1495	0.2825	0.2889
Ejecución 3	0.2374	0.2749	0.3043	0.1634	0.2824	0.2621	0.2994	0.1749
Ejecución 4	-0.0631	0.2703	0.2839	0.2705	0.2836	0.2778	0.2703	0.2675
Ejecución 5	-0.0594	0.2454	0.2984	0.2651	0.2978	0.2405	0.2568	0.2468
Ejecución 6	-0.0781	0.2137	0.2546	0.2504	0.2015	0.2515	0.2481	0.2641
Ejecución 7	-0.075	0.2875	0.25	0.3007	0.2084	0.2015	0.2648	0.2108
Ejecución 8	-0.0731	0.3015	0.2978	0.2607	0.3105	0.2098	0.248	0.2504
Ejecución 9	-0.0694	0.2981	0.2034	0.2189	0.2554	0.23	0.2648	0.268

Continúa en la siguiente página

Tabla A.2 – continúa de la página anterior								
Nombre	0% restricciones		10% restricciones		15% restricciones		20% restricciones	
	SOM zD	SOM zD	SOM zD	SOM zD	SOM zD	SOM zD	SOM zD	SOM zD
Ejecución 10	-0.0697	0.2648	0.3105	0.2085	0.2645	0.2001	0.2354	0.1907
Pima	0.0461 ± 0.0029	0.0328 ± 0.0334	0.0753 ± 0.0426	0.0701 ± 0.0276	0.1041 ± 0.0401	0.0897 ± 0.0178	0.1227 ± 0.048	0.0725 ± 0.0241
Ejecución 1	0.0464	0.0695	0.0432	0.0963	0.1146	0.1148	0.0461	0.0837
Ejecución 2	0.0503	-0.0005	0.0301	0.0239	0.0477	0.0973	0.1197	0.097
Ejecución 3	0.0421	-0.0006	0.1386	0.0832	0.1593	0.0768	0.1528	0.0327
Ejecución 4	0.0454	0.0626	0.0891	0.0768	0.0947	0.0695	0.172	0.0645
Ejecución 5	0.0405	0.0315	0.0897	0.0617	0.1504	0.0847	0.1205	0.078
Ejecución 6	0.0391	0.0297	0.0456	0.04	0.1978	0.0548	0.112	0.087
Ejecución 7	0.0481	0.0384	0.0751	0.0834	0.0978	0.1109	0.1058	0.0654
Ejecución 8	0.0495	0.0415	0.0945	0.0392	0.0645	0.024	0.098	0.0704
Ejecución 9	0.043	0.0297	0.0604	0.0649	0.0124	0.0678	0.0854	0.0841
Ejecución 10	0.0458	0.0317	0.0537	0.0897	0.0345	0.0941	0.0705	0.0702
Saheart	-0.0167 ± 0.015	-0.0039 ± 0.0077	0.068 ± 0.0047	0.0752 ± 0.0086	0.0865 ± 0.0144	0.0787 ± 0.0088	0.069 ± 0.0154	0.0777 ± 0.0125
Ejecución 1	-0.0236	-0.0072	0.0617	0.0744	0.0617	0.0657	0.0431	0.0905
Ejecución 2	0.0089	-0.0114	0.072	0.0884	0.0923	0.0849	0.0786	0.0602
Ejecución 3	-0.0225	0.009	0.073	0.0735	0.0974	0.0884	0.0824	0.0883
Ejecución 4	-0.0294	-0.006	0.0653	0.0643	0.0943	0.0756	0.0718	0.0716
Ejecución 5	-0.0193	-0.0059	0.0754	0.0532	0.0874	0.8184	0.1202	0.078
Ejecución 6	-0.012	-0.0049	0.0546	0.0863	0.0745	0.7054	0.1332	0.0815
Ejecución 7	-0.0098	-0.0082	0.0416	0.0749	0.0914	0.615	0.1405	0.079
Ejecución 8	-0.0204	0.0089	0.0318	0.0804	0.0924	0.7932	0.1508	0.0814
Ejecución 9	-0.0189	-0.0092	0.0984	0.0562	0.0784	0.7901	0.1302	0.079
Ejecución 10	-0.013	0.0092	0.0731	0.0654	0.0987	0.8214	0.1102	0.072
Satimage	0.3753 ± 0.0161	0.4815 ± 0.035	0.3885 ± 0.0233	0.4908 ± 0.0525	0.3611 ± 0.0284	0.4464 ± 0.011	0.3744 ± 0.0286	0.4717 ± 0.0365
Ejecución 1	0.3488	0.4545	0.3575	0.435	0.353	0.4477	0.4231	0.4497
Ejecución 2	0.3819	0.5411	0.4095	0.5701	0.409	0.4283	0.3661	0.4424
Ejecución 3	0.3923	0.4717	0.3743	0.4659	0.3474	0.4574	0.3526	0.5339
Ejecución 4	0.3779	0.4585	0.4124	0.486	0.3349	0.4519	0.3558	0.4603
Ejecución 5	0.37	0.4784	0.3903	0.3853	0.3545	0.4406	0.3807	0.4802
Ejecución 6	0.337	0.4312	0.4127	0.392	0.3709	0.5168	0.3074	0.4054
Ejecución 7	0.3364	0.4901	0.3815	0.5192	0.4105	0.5245	0.3604	0.4138
Ejecución 8	0.2981	0.452	0.3715	0.5073	0.2978	0.425	0.3504	0.4801
Ejecución 9	0.412	0.3907	0.313	0.4509	0.3409	0.4538	0.3608	0.4608
Ejecución 10	0.3807	0.5233	0.4155	0.4928	0.3881	0.452	0.3804	0.4728
Segment	0.2562 ± 0.0129	0.2706 ± 0.0141	0.2683 ± 0.0232	0.2747 ± 0.0177	0.2752 ± 0.0138	0.2868 ± 0.0517	0.2782 ± 0.0278	0.2806 ± 0.0288
Ejecución 1	0.2761	0.2917	0.2576	0.2824	0.2584	0.2552	0.2585	0.3253
Ejecución 2	0.2482	0.2639	0.2595	0.291	0.273	0.3431	0.2475	0.2801
Ejecución 3	0.2587	0.2533	0.3078	0.2447	0.2969	0.3301	0.2875	0.2714
Ejecución 4	0.2419	0.2731	0.2482	0.2805	0.2722	0.2185	0.3192	0.2453
Ejecución 5	0.264	0.2984	0.2545	0.2698	0.2978	0.2915	0.2704	0.2907
Ejecución 6	0.2304	0.3105	0.2301	0.2654	0.215	0.3412	0.2508	0.3018
Ejecución 7	0.3012	0.2548	0.2915	0.3191	0.3105	0.242	0.298	0.2708
Ejecución 8	0.2405	0.2745	0.2018	0.2856	0.3547	0.2384	0.3015	0.2804
Ejecución 9	0.3105	0.1904	0.3012	0.2564	0.3154	0.2812	0.2045	0.3021
Ejecución 10	0.2546	0.3048	0.2046	0.2951	0.2456	0.315	0.2701	0.2801
Sonar	-0.0044 ± 0.0003	-0.003 ± 0.0024	0.0139 ± 0.0164	0.0081 ± 0.0114	0.008 ± 0.0066	-0.0003 ± 0.0065	0.0171 ± 0.0127	0.0284 ± 0.0188
Ejecución 1	-0.0043	0.0011	-0.0042	0.0108	-0.0033	-0.0033	-0.0024	-0.0033
Ejecución 2	-0.0046	0.0064	0.0064	0.0252	0.0133	0.0108	0.022	0.0401
Ejecución 3	-0.0038	-0.0039	0.0133	-0.0048	0.0108	-0.0039	0.0162	0.0444
Ejecución 4	-0.0046	0.0041	0.0401	0.0011	0.0108	-0.0046	0.0325	0.0323
Ejecución 5	0.004	0.0015	0.0124	0.0094	0.0087	0.0003	0.0167	0.0301
Ejecución 6	0.0039	0.002	0.0021	0.0092	0.0045	0.0001	0.0148	0.027
Ejecución 7	0.004	-0.0032	0.0915	0.0052	0.0067	0.0002	0.0193	0.0298
Ejecución 8	0.0039	-0.0014	0.0164	-0.0385	0.0094	0.0006	0.0221	0.0188
Ejecución 9	0.0038	0.0056	0.0137	0.0087	0.0104	0.0001	0.0178	0.0298
Ejecución 10	0.0045	0.0025	0.0212	0.0094	0.0005	0.0011	0.0152	0.0218
Soybean	0.5842 ± 0.0256	0.5414 ± 0.0343	0.5846 ± 0.019	0.5472 ± 0.0124	0.6236 ± 0.0787	0.5215 ± 0.0767	0.5085 ± 0.0598	0.5538 ± 0.0197
Ejecución 1	0.6097	0.5814	0.5814	0.5634	0.6349	0.3899	0.5063	0.5814
Ejecución 2	0.5562	0.4916	0.5567	0.5289	0.5485	0.5512	0.4216	0.5351
Ejecución 3	0.561	0.5634	0.5904	0.5451	0.5634	0.5814	0.5904	0.5634
Ejecución 4	0.6097	0.5289	0.6097	0.5512	0.7477	0.5634	0.5156	0.5351
Ejecución 5	0.5804	0.6154	0.5903	0.2871	0.6898	0.5406	0.5048	0.5401
Ejecución 6	0.6011	0.5978	0.6134	0.6045	0.6784	0.5102	0.4908	0.531
Ejecución 7	0.554	0.5015	0.751	0.4909	0.5974	0.5021	0.4804	0.5408
Ejecución 8	0.568	0.5712	0.5301	0.5188	0.5301	0.4912	0.517	0.5707
Ejecución 9	0.5824	0.5354	0.584	0.5616	0.637	0.4823	0.5208	0.5087
Ejecución 10	0.5971	0.5504	0.5143	0.1617	0.684	0.5304	0.4705	0.4908
Spambase	0.0473 ± 0.0038	0.0734 ± 0.0008	0.1642 ± 0.0144	0.1493 ± 0.005	0.153 ± 0.0027	0.1545 ± 0.0059	0.158 ± 0.0101	0.1526 ± 0.0051
Ejecución 1	0.0517	0.0743	0.1529	0.1448	0.1508	0.1508	0.1456	0.1519
Ejecución 2	0.0476	0.0738	0.1547	0.1501	0.1501	0.1495	0.1587	0.1498
Ejecución 3	0.0486	0.0722	0.1885	0.145	0.1539	0.1645	0.1733	0.161
Ejecución 4	0.0412	0.0731	0.1605	0.1571	0.1569	0.1531	0.1541	0.1475
Ejecución 5	0.0433	0.0897	0.1738	0.1261	0.145	0.167	0.1604	0.1648
Ejecución 6	0.0485	0.0789	0.1405	0.1871	0.1846	0.197	0.1708	0.142
Ejecución 7	0.0518	0.0697	0.182	0.1308	0.167	0.1887	0.1405	0.154
Ejecución 8	0.0435	0.0752	0.173	0.122	0.0978	0.1403	0.153	0.112
Ejecución 9	0.0468	0.0615	0.1981	0.1693	0.1845	0.138	0.1309	0.1604
Ejecución 10	0.045	0.0801	0.1349	0.1065	0.2154	0.1645	0.168	0.1538
Spectfheart	-0.0974 ± 0.0045	-0.1056 ± 0.0009	0.294 ± 0.037	0.2542 ± 0.0377	0.2149 ± 0.0174	0.2921 ± 0.0093	0.2568 ± 0.0174	0.2787 ± 0.0341
Ejecución 1	-0.1034	-0.1061	0.2985	0.2345	0.2262	0.2771	0.2809	0.309
Ejecución 2	-0.0983	-0.1046	0.2476	0.2035	0.2346	0.3024	0.264	0.2261
Ejecución 3	-0.097	-0.1066	0.2799	0.3004	0.2093	0.294	0.2343	0.2707
Ejecución 4	-0.0908	-0.1047	0.3498	0.2785	0.1891	0.2949	0.2479	0.3088
Ejecución 5	-0.098	-0.0981	0.3015	0.2065	0.2654	0.2801	0.2604	0.2807
Ejecución 6	-0.1055	-0.0994	0.2854	0.2248	0.2015	0.2708	0.2304	0.3078
Ejecución 7	-0.0874	-0.112	0.2401	0.2647	0.2981	0.3105	0.24	0.3107
Ejecución 8	-0.0831	-0.1042	0.3382	0.2378	0.2105	0.3201	0.2105	0.2604
Ejecución 9	-0.1205	-0.0894	0.2901	0.2907	0.1972	0.3083	0.2304	0.2738
Ejecución 10	-0.0983	-0.1033	0.3172	0.2774	0.1725	0.2705	0.2105	0.271
Tae	0.008 ± 0.0059	0.0008 ± 0.0099	0.019 ± 0.0101	0.0484 ± 0.0178	0.0314 ± 0.0242	0.0283 ± 0.0126	0.0252 ± 0.0117	0.0427 ± 0.0206
Ejecución 1	0.0179	-0.0072	0.019	0.0703	-0.0002	0.0488	0.0216	0.0137

Continúa en la siguiente página

A. Resultados experimentales completos.

Tabla A.2 – continúa de la página anterior

Nombre	0 % restricciones		10 % restricciones		15 % restricciones		20 % restricciones	
	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D
Ejecución 2	0.0042	-0.0033	0.0334	0.043	0.0199	0.0147	0.0393	0.0542
Ejecución 3	0.007	0.0178	0.0184	0.0224	0.0648	0.0238	0.0315	0.0343
Ejecución 4	0.0028	-0.0038	0.005	0.0576	0.0411	0.0257	0.0081	0.0683
Ejecución 5	0.0082	0.0007	0.0201	0.0471	0.0544	0.029	0.026	0.0458
Ejecución 6	0.0081	0.0009	0.0184	0.032	0.0315	0.0314	0.024	0.039
Ejecución 7	0.0079	-0.008	0.0197	0.0587	0.0245	0.025	0.019	0.0431
Ejecución 8	0.0075	-0.0047	0.0154	0.0408	0.0297	0.019	0.027	0.048
Ejecución 9	0.0089	0.0003	0.0213	0.0421	0.0247	0.035	0.019	0.0431
Ejecución 10	0.0108	0.0009	0.0207	0.0398	0.0412	0.028	0.0287	0.0401
Thyroid	-0.0138 ± 0.0124	-0.0118 ± 0.0056	-0.0247 ± 0.002	-0.0177 ± 0.0116	-0.0104 ± 0.0257	-0.0232 ± 0.0015	-0.0185 ± 0.0116	-0.0245 ± 0.002
Ejecución 1	0.001	-0.0084	-0.0258	-0.0231	0.0339	-0.0209	-0.0304	-0.0278
Ejecución 2	-0.011	-0.0086	-0.0274	0.0023	-0.0266	-0.0242	0.0006	-0.0242
Ejecución 3	-0.0118	-0.0086	-0.0223	-0.0235	-0.0225	-0.0226	-0.021	-0.0234
Ejecución 4	-0.0333	-0.0215	-0.0231	-0.0265	-0.0264	-0.0249	-0.0233	-0.0226
Ejecución 5	-0.0105	-0.0135	-0.0249	0.0019	-0.0154	-0.0225	-0.0198	-0.0235
Ejecución 6	-0.0189	-0.0164	-0.0214	0.0022	-0.0948	-0.0315	-0.0212	-0.0242
Ejecución 7	-0.0131	-0.0241	-0.0345	-0.0165	0.0327	-0.0154	-0.0197	-0.0198
Ejecución 8	-0.0165	-0.0164	-0.0201	-0.0197	-0.0146	-0.0234	-0.0154	-0.0321
Ejecución 9	-0.015	-0.0345	-0.0197	0.0019	-0.0121	-0.0215	-0.0191	-0.0245
Ejecución 10	-0.014	-0.0154	-0.0284	-0.1474	-0.0135	-0.0241	-0.0182	-0.0254
Titanic	0.0781 ± 0.0889	0.1384 ± 0.0773	0.2209 ± 0.0052	0.225 ± 0.0095	0.161 ± 0.095	0.2271 ± 0.007	0.1595 ± 0.0941	0.225 ± 0.0095
Ejecución 1	-0.0034	0.2096	0.2158	0.2248	0.2158	0.2271	0.2158	0.2309
Ejecución 2	0.1095	0.1095	0.2248	0.2309	0.2158	0.2158	0.2096	0.2345
Ejecución 3	0.2096	0.2096	0.2271	0.2345	-0.0034	0.2345	0.2158	0.2248
Ejecución 4	-0.0034	0.0248	0.2158	0.2096	0.2158	0.2309	-0.0034	0.2096
Ejecución 5	0.0321	0.1245	0.2014	0.2177	0.0154	0.2105	0.1604	0.2308
Ejecución 6	0.0642	0.1354	0.2318	0.2446	0.0124	0.2348	0.1708	0.2405
Ejecución 7	0.089	0.1055	0.2214	0.2065	0.0167	0.2705	0.1543	0.208
Ejecución 8	-0.0035	0.1504	0.2972	0.2676	0.0197	0.1908	0.1972	0.2308
Ejecución 9	-0.0029	0.164	0.2014	0.2304	0.0157	0.205	0.1024	0.2108
Ejecución 10	0.0102	0.0978	0.1973	0.2774	0.0094	0.2605	0.1305	0.245
Vehicle	0.1033 ± 0.0078	0.0773 ± 0.0099	0.1379 ± 0.011	0.0919 ± 0.0057	0.121 ± 0.0275	0.1024 ± 0.0296	0.1292 ± 0.0121	0.0848 ± 0.0068
Ejecución 1	-0.1026	0.0931	0.1289	0.101	0.142	0.0875	0.1159	0.0899
Ejecución 2	0.0986	0.0666	0.125	0.0881	0.0912	0.0882	0.1331	0.0877
Ejecución 3	0.1162	0.0774	0.1477	0.0923	0.1541	0.0803	0.1471	0.0884
Ejecución 4	0.0958	0.0718	0.1497	0.086	0.0967	0.1534	0.1206	0.073
Ejecución 5	0.098	0.0567	0.1543	0.1026	0.1201	0.0904	0.1304	0.0794
Ejecución 6	0.075	0.0784	0.1242	0.0761	0.1197	0.1204	0.1204	0.084
Ejecución 7	0.1502	0.0845	0.1024	0.0546	0.0918	0.1102	0.1058	0.068
Ejecución 8	0.1054	0.0731	0.1872	0.1026	0.0931	0.0918	0.0918	0.087
Ejecución 9	0.1098	0.0721	0.164	0.0969	0.1348	0.0891	0.2105	0.0814
Ejecución 10	0.09	0.0845	0.1023	0.0741	0.1548	0.083	0.164	0.0759
Wdbc	0.3038 ± 0.0246	0.3614 ± 0.0356	0.6382 ± 0.0177	0.6077 ± 0.0163	0.619 ± 0.0118	0.6153 ± 0.0264	0.6376 ± 0.0189	0.6122 ± 0.0145
Ejecución 1	0.2958	0.331	0.6098	0.5994	0.6344	0.5939	0.629	0.6218
Ejecución 2	0.2958	0.4057	0.6536	0.6098	0.6115	0.6234	0.6254	0.6156
Ejecución 3	0.2787	0.3865	0.6367	0.5887	0.6256	0.5887	0.6701	0.6238
Ejecución 4	0.3446	0.3221	0.6525	0.6328	0.6042	0.6548	0.6256	0.5876
Ejecución 5	0.3015	0.3757	0.6154	0.8045	0.6254	0.7055	0.7048	0.6481
Ejecución 6	0.298	0.3451	0.6248	0.7456	0.5948	0.7309	0.5096	0.615
Ejecución 7	0.2804	0.2948	0.6015	0.4667	0.5715	0.5048	0.634	0.6021
Ejecución 8	0.2704	0.3154	0.6345	0.8054	0.5645	0.6015	0.6438	0.6319
Ejecución 9	0.3014	0.3845	0.6245	0.692	0.6289	0.6278	0.5401	0.598
Ejecución 10	0.3201	0.3201	0.6548	0.6006	0.6424	0.602	0.7185	0.5907
Wine	0.2438 ± 0.0511	0.3453 ± 0.0123	0.3934 ± 0.0245	0.3885 ± 0.0132	0.3989 ± 0.0053	0.4213 ± 0.0258	0.4299 ± 0.0102	0.4017 ± 0.0248
Ejecución 1	0.196	0.3504	0.3629	0.3748	0.3981	0.4539	0.433	0.3609
Ejecución 2	0.2549	0.3374	0.4134	0.3814	0.4074	0.3856	0.4275	0.4136
Ejecución 3	0.3228	0.3626	0.376	0.3879	0.3928	0.4101	0.4154	0.4269
Ejecución 4	0.2015	0.3305	0.4212	0.4099	0.397	0.4354	0.4437	0.4053
Ejecución 5	0.254	0.3504	0.4024	0.3795	0.4105	0.4305	0.415	0.4015
Ejecución 6	0.264	0.3345	0.4315	0.3146	0.3945	0.4915	0.4125	0.4902
Ejecución 7	0.2301	0.3845	0.3871	0.4014	0.3728	0.3904	0.394	0.4282
Ejecución 8	0.2012	0.382	0.3648	0.3785	0.3452	0.337	0.4605	0.4389
Ejecución 9	0.265	0.4015	0.3001	0.3621	0.4321	0.318	0.3108	0.4715
Ejecución 10	0.2405	0.324	0.3978	0.381	0.3514	0.4502	0.3508	0.4087
Zoo	0.674 ± 0.0172	0.6338 ± 0.0172	0.6642 ± 0.0216	0.6484 ± 0.0146	0.6899 ± 0.095	0.7551 ± 0.0153	0.8103 ± 0.0018	0.7157 ± 0.0837
Ejecución 1	0.6874	0.6332	0.685	0.6587	0.854	0.7551	0.8131	0.7564
Ejecución 2	0.6452	0.6587	0.6718	0.626	0.644	0.7762	0.8093	0.7564
Ejecución 3	0.6763	0.61	0.6279	0.645	0.6251	0.756	0.8083	0.7784
Ejecución 4	0.6869	0.6332	0.6718	0.6638	0.6362	0.733	0.8104	0.5715
Ejecución 5	0.6154	0.6254	0.667	0.6563	0.7504	0.7468	0.8122	0.6751
Ejecución 6	0.6904	0.6045	0.6319	0.6761	0.6154	0.7954	0.9804	0.774
Ejecución 7	0.675	0.6884	0.6048	0.7165	0.721	0.7307	0.7504	0.6901
Ejecución 8	0.6315	0.5971	0.6708	0.6109	0.6598	0.781	0.8407	0.672
Ejecución 9	0.6874	0.6108	0.6943	0.642	0.6301	0.6904	0.8316	0.7597
Ejecución 10	0.6915	0.6208	0.6821	0.6305	0.6815	0.7201	0.8112	0.5408
Media	0.1668 ± 0.2113	0.1859 ± 0.2158	0.264 ± 0.2111	0.2633 ± 0.206	0.2746 ± 0.22	0.2798 ± 0.2172	0.2732 ± 0.2186	0.2701 ± 0.2158

Tabla A.3.: Resultados de las ejecuciones del CopSoft-SOM.

Nombre	0% restricciones		10% restricciones		15% restricciones		20% restricciones	
	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D
Appendicitis	0.5164 ± 0.0247	0.4975 ± 0.0343	0.2331 ± 0.2024	0.0809 ± 0.1275	0.5425 ± 0.0573	0.5626 ± 0.0981	0.5524 ± 0.1083	0.459 ± 0.1134
Ejecución 1	0.5153	0.5256	-0.0599	-0.0075	0.6041	0.7114	0.3925	0.3254
Ejecución 2	0.4746	0.5256	0.4688	0.063	0.5779	0.3813	0.4423	0.4316
Ejecución 3	0.5153	0.4862	0.3141	-0.0315	0.5227	0.6365	0.504	0.4862
Ejecución 4	0.5153	0.4199	0.0956	0.3506	0.5673	0.5779	0.4619	0.4746
Ejecución 5	0.5459	0.4966	0.4688	-0.0218	0.4519	0.6631	0.5875	0.5962
Ejecución 6	0.5153	0.5256	0.2531	0.0394	0.4322	0.5508	0.6878	0.3813
Ejecución 7	0.5459	0.5256	0.5559	-0.0218	0.5673	0.4068	0.6495	0.4466
Ejecución 8	0.4746	0.4583	0.0932	0.2664	0.5227	0.5875	0.629	0.4108
Ejecución 9	0.5459	0.5256	-0.0014	0.0094	0.5673	0.5459	0.7114	0.7114
Ejecución 10	0.5153	0.4862	0.1425	0.1623	0.6114	0.5649	0.4583	0.3254
Balance	0.0607 ± 0.0177	0.0578 ± 0.0312	0.4668 ± 0.0486	0.4866 ± 0.039	0.5149 ± 0.0378	0.5204 ± 0.0478	0.5486 ± 0.0879	0.5946 ± 0.0357
Ejecución 1	0.06	0.0168	0.4831	0.4875	0.5156	0.4377	0.4998	0.5897
Ejecución 2	0.0628	0.0903	0.4556	0.4475	0.4829	0.5448	0.5423	0.562
Ejecución 3	0.0618	0.094	0.4436	0.5173	0.5618	0.5909	0.5891	0.5449
Ejecución 4	0.0481	0.0614	0.4348	0.5498	0.5118	0.5189	0.6064	0.6245
Ejecución 5	0.0604	0.0723	0.5718	0.477	0.4903	0.5513	0.6557	0.5926
Ejecución 6	0.0746	0.0112	0.4109	0.4812	0.5213	0.5702	0.6041	0.6339
Ejecución 7	0.0261	0.0326	0.4039	0.3986	0.5141	0.5412	0.6293	0.626
Ejecución 8	0.1002	0.094	0.4729	0.506	0.482	0.52	0.3376	0.6013
Ejecución 9	0.0559	0.0779	0.466	0.5063	0.5997	0.4514	0.485	0.6384
Ejecución 10	0.0572	0.0273	0.5257	0.495	0.4691	0.4776	0.5362	0.5328
Banana	0.003 ± 0.0036	0.0437 ± 0.0306	0.2805 ± 0.0332	0.4254 ± 0.1078	0.5493 ± 0.0538	0.5567 ± 0.1134	0.6192 ± 0.1082	0.6486 ± 0.1009
Ejecución 1	0.0083	0.009	0.2618	0.4505	0.5152	0.5113	0.6619	0.5138
Ejecución 2	0.0037	0.0698	0.2724	0.4955	0.5672	0.4144	0.7078	0.8064
Ejecución 3	0.0012	0.0842	0.2817	0.508	0.5767	0.7121	0.7684	0.5636
Ejecución 4	0.003	0.0137	0.3334	0.4237	0.5748	0.4506	0.6931	0.6001
Ejecución 5	0.0024	0.0203	0.2659	0.4538	0.5786	0.3749	0.6952	0.8042
Ejecución 6	0.0004	0.0755	0.3364	0.4387	0.6276	0.6436	0.5998	0.7143
Ejecución 7	0.0023	0.0185	0.3035	0.115	0.4157	0.5962	0.4525	0.5112
Ejecución 8	-0.005	0.0247	0.2289	0.4901	0.5277	0.6744	0.4241	0.6724
Ejecución 9	0.0057	0.0304	0.2763	0.4074	0.5729	0.5172	0.5375	0.6745
Ejecución 10	0.0075	0.0903	0.2452	0.471	0.5369	0.6723	0.6516	0.6257
Breast_cancer	0.2824 ± 0.0212	0.3576 ± 0.0122	0.7298 ± 0.0315	0.7669 ± 0.0365	0.755 ± 0.0277	0.8026 ± 0.0246	0.7974 ± 0.0299	0.822 ± 0.0264
Ejecución 1	0.2537	0.3631	0.7122	0.7734	0.7866	0.7859	0.8116	0.8569
Ejecución 2	0.3089	0.3538	0.7238	0.8243	0.7305	0.8437	0.7801	0.8244
Ejecución 3	0.2872	0.3447	0.7373	0.7545	0.7736	0.8115	0.8179	0.818
Ejecución 4	0.2787	0.3724	0.6954	0.7672	0.7195	0.8115	0.7373	0.8504
Ejecución 5	0.2456	0.3447	0.7553	0.805	0.7681	0.8177	0.7989	0.8438
Ejecución 6	0.3045	0.3771	0.805	0.7236	0.7375	0.7549	0.8437	0.7734
Ejecución 7	0.2703	0.3401	0.7072	0.7302	0.7484	0.7797	0.7559	0.7861
Ejecución 8	0.3045	0.3492	0.7487	0.755	0.8056	0.7923	0.8056	0.8051
Ejecución 9	0.3002	0.3681	0.7119	0.7178	0.7616	0.8306	0.8054	0.8179
Ejecución 10	0.2703	0.3631	0.701	0.8179	0.7184	0.7987	0.8179	0.8439
Contraceptive	0.0118 ± 0.001	0.022 ± 0.0011	0.1226 ± 0.0224	0.1671 ± 0.016	0.4906 ± 0.0318	0.5147 ± 0.0453	0.6234 ± 0.0396	0.6892 ± 0.0271
Ejecución 1	0.0102	0.0236	0.1229	0.1503	0.5474	0.5555	0.6203	0.7296
Ejecución 2	0.0128	0.0197	0.1187	0.1483	0.5214	0.5567	0.5754	0.7157
Ejecución 3	0.0116	0.0219	0.1097	0.1827	0.4597	0.4831	0.6568	0.6979
Ejecución 4	0.0117	0.023	0.1655	0.1481	0.5006	0.4812	0.6119	0.7097
Ejecución 5	0.014	0.0221	0.1303	0.1953	0.4593	0.6005	0.6972	0.6695
Ejecución 6	0.0115	0.0221	0.1005	0.1662	0.438	0.471	0.6429	0.6554
Ejecución 7	0.0112	0.0211	0.1512	0.1587	0.4949	0.5365	0.6485	0.6686
Ejecución 8	0.0109	0.0233	0.1342	0.1734	0.472	0.5137	0.549	0.6881
Ejecución 9	0.0122	0.022	0.1031	0.1867	0.5182	0.4426	0.6216	0.6445
Ejecución 10	0.0118	0.0208	0.0895	0.1614	0.4942	0.5063	0.6108	0.7132
Ecoli	0.2111 ± 0.0083	0.2989 ± 0.0142	0.3373 ± 0.0328	0.2952 ± 0.031	0.5178 ± 0.0542	0.4706 ± 0.0556	0.4985 ± 0.0476	0.5612 ± 0.0615
Ejecución 1	0.2147	0.3121	0.3353	0.2673	0.5711	0.3852	0.4524	0.5325
Ejecución 2	0.2095	0.3088	0.3224	0.3357	0.5267	0.3791	0.4838	0.6384
Ejecución 3	0.2166	0.2845	0.3156	0.2648	0.4095	0.4881	0.5467	0.5653
Ejecución 4	0.2126	0.2967	0.3482	0.2867	0.5879	0.4973	0.4478	0.4737
Ejecución 5	0.2043	0.2806	0.3606	0.3364	0.4611	0.486	0.5469	0.5288
Ejecución 6	0.2253	0.2945	0.3549	0.2538	0.4801	0.4279	0.5827	0.452
Ejecución 7	0.2019	0.2871	0.4119	0.3254	0.5147	0.4569	0.5326	0.6363
Ejecución 8	0.1952	0.3044	0.3256	0.2592	0.5497	0.5363	0.4372	0.6241
Ejecución 9	0.2181	0.3297	0.3154	0.3062	0.5787	0.5546	0.4884	0.5752
Ejecución 10	0.2122	0.2907	0.2832	0.1612	0.4982	0.4942	0.4669	0.5857
Glass	0.213 ± 0.0202	0.2307 ± 0.0171	0.138 ± 0.036	0.1327 ± 0.032	0.2135 ± 0.0412	0.2532 ± 0.0195	0.2785 ± 0.0398	0.2943 ± 0.0582
Ejecución 1	0.2121	0.2587	0.116	0.1255	0.2635	0.2437	0.2516	0.3567
Ejecución 2	0.2039	0.2137	0.0915	0.1771	0.2927	0.2274	0.2664	0.3033
Ejecución 3	0.1803	0.2301	0.1593	0.1275	0.2132	0.2463	0.3515	0.1806
Ejecución 4	0.2361	0.2063	0.1523	0.1197	0.1776	0.2534	0.2132	0.2432
Ejecución 5	0.2365	0.2524	0.1754	0.1775	0.1913	0.2683	0.2769	0.3667
Ejecución 6	0.1934	0.2384	0.1372	0.0806	0.1742	0.2502	0.2745	0.3061
Ejecución 7	0.2154	0.2178	0.0962	0.1477	0.241	0.233	0.306	0.3683
Ejecución 8	0.2247	0.2159	0.1329	0.1519	0.1552	0.2457	0.2457	0.2408
Ejecución 9	0.188	0.248	0.2123	0.1386	0.2318	0.2997	0.263	0.2707
Ejecución 10	0.2391	0.2258	0.1068	0.0811	0.1943	0.2643	0.336	0.3063
Haberman	-0.0136 ± 0.0109	-0.0247 ± 0.0106	0.3254 ± 0.0376	0.3297 ± 0.0614	0.3851 ± 0.0567	0.3942 ± 0.0523	0.3576 ± 0.0597	0.465 ± 0.055
Ejecución 1	-0.0086	-0.0277	0.3047	0.3245	0.3484	0.3411	0.3982	0.4393
Ejecución 2	-0.007	-0.0123	0.3282	0.2851	0.447	0.4186	0.3579	0.3857
Ejecución 3	-0.0068	-0.0343	0.3501	0.2561	0.3081	0.3765	0.3381	0.4969
Ejecución 4	-0.0165	-0.0107	0.3084	0.276	0.3908	0.464	0.2548	0.5224
Ejecución 5	-0.0145	-0.0107	0.2542	0.3317	0.39	0.4084	0.2985	0.4441
Ejecución 6	-0.0127	-0.0329	0.3028	0.2768	0.3881	0.4646	0.4087	0.5237
Ejecución 7	-0.0437	-0.0282	0.3819	0.4186	0.318	0.3365	0.3026	0.5344
Ejecución 8	-0.0165	-0.0369	0.3231	0.3208	0.5104	0.3506	0.4657	0.3649
Ejecución 9	-0.003	-0.0155	0.3874	0.3502	0.3924	0.4558	0.3476	0.4601
Ejecución 10	-0.007	-0.0372	0.3134	0.4568	0.3577	0.3262	0.4037	0.4787
Hayes_roth	0.0802 ± 0.0223	0.0224 ± 0.0285	0.0359 ± 0.036	0.0267 ± 0.0275	0.1354 ± 0.0742	0.1346 ± 0.0438	0.1419 ± 0.0447	0.1277 ± 0.0784

Continúa en la siguiente página

A. Resultados experimentales completos.

Tabla A.3 – continúa de la página anterior

Nombre	0 % restricciones		10 % restricciones		15 % restricciones		20 % restricciones	
	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D
Ejecución 1	0.1334	0.0533	0.0607	0.0048	0.2672	0.1055	0.0821	0.1057
Ejecución 2	0.1007	0.0162	0.0664	0.0064	0.0697	0.2083	0.1454	0.0395
Ejecución 3	0.0902	-0.0131	-0.0101	0.0655	0.2277	0.0917	0.2334	0.0425
Ejecución 4	0.0603	0.0476	-0.0019	0.0329	0.0733	0.1517	0.1127	0.2123
Ejecución 5	0.0732	0.0043	0.0122	-0.0032	0.0758	0.155	0.1392	0.2409
Ejecución 6	0.0693	0.0193	0.0358	0.077	0.151	0.05	0.1179	0.0527
Ejecución 7	0.0692	0.0706	0.0076	0.018	0.123	0.1875	0.127	0.0668
Ejecución 8	0.0778	0.0442	0.0778	-0.0088	0.224	0.1412	0.203	0.0922
Ejecución 9	0.0784	-0.0166	0.0999	0.0453	0.0937	0.1396	0.1632	0.234
Ejecución 10	0.0495	-0.0016	0.0102	0.0296	0.0488	0.1157	0.0947	0.191
Heart	0.0033 ± 0.0034	0.0085 ± 0.0036	0.1509 ± 0.0512	0.0976 ± 0.0745	0.136 ± 0.0728	0.1623 ± 0.0393	0.1981 ± 0.044	0.2311 ± 0.019
Ejecución 1	0.0034	0.0116	0.169	0.0981	0.2216	0.1945	0.2145	0.2208
Ejecución 2	0.0005	0.0074	0.1285	0.0042	0.0396	0.151	0.2358	0.2505
Ejecución 3	-0.0008	0.0131	0.1335	0.093	0.1624	0.117	0.1285	0.2079
Ejecución 4	0.0033	0.0028	0.1506	0.1879	0.1568	0.1752	0.1813	0.1934
Ejecución 5	0.0049	0.0094	0.0492	0.1173	0.2012	0.2012	0.2739	0.251
Ejecución 6	0.0009	0.0134	0.151	0.2424	-0.0038	0.084	0.1285	0.2434
Ejecución 7	-0.0006	0.0094	0.1282	0.0152	0.188	0.1222	0.2352	0.2219
Ejecución 8	0.0071	0.0096	0.2585	-0.0026	0.0979	0.188	0.2069	0.2435
Ejecución 9	0.0105	0.0041	0.1392	0.1171	0.0884	0.2012	0.2011	0.2501
Ejecución 10	0.0036	0.0042	0.2012	0.1029	0.2074	0.188	0.1752	0.229
Ionosphere	0.0106 ± 0.0894	0.0019 ± 0.0192	0.3568 ± 0.1386	0.3059 ± 0.0354	0.5524 ± 0.0619	0.471 ± 0.0726	0.6364 ± 0.0383	0.5948 ± 0.06
Ejecución 1	-0.0104	0.0438	0.5628	0.2755	0.597	0.369	0.692	0.5509
Ejecución 2	-0.0329	-0.0077	0.2589	0.2877	0.5001	0.5428	0.6072	0.5517
Ejecución 3	-0.017	0.0241	0.4205	0.3187	0.5344	0.4579	0.6464	0.6156
Ejecución 4	-0.0074	-0.0211	0.3283	0.3005	0.5974	0.4731	0.6441	0.624
Ejecución 5	-0.0083	0.0006	0.1724	0.3126	0.5521	0.5063	0.7116	0.6913
Ejecución 6	0.2768	0.0115	0.1926	0.4009	0.5909	0.3519	0.6443	0.5689
Ejecución 7	-0.0074	-0.0019	0.4518	0.2941	0.5521	0.5242	0.6069	0.6717
Ejecución 8	-0.0153	-0.0136	0.4756	0.3002	0.5977	0.5242	0.5882	0.515
Ejecución 9	-0.033	0.0029	0.5126	0.2632	0.3953	0.5689	0.6252	0.5155
Ejecución 10	-0.039	-0.0194	0.1927	0.3055	0.6069	0.3914	0.5977	0.6435
Iris	0.6319 ± 0.0697	0.7542 ± 0.0716	0.4354 ± 0.0716	0.4322 ± 0.0756	0.8573 ± 0.0759	0.8511 ± 0.0762	0.9136 ± 0.0912	0.9512 ± 0.0367
Ejecución 1	0.7196	0.688	0.4208	0.3178	0.8422	0.9214	0.9799	0.9799
Ejecución 2	0.5496	0.8343	0.4241	0.5819	0.941	0.8255	0.7592	0.9799
Ejecución 3	0.5587	0.8341	0.4208	0.4531	0.9002	0.8841	0.9799	0.9222
Ejecución 4	0.7196	0.6643	0.4545	0.5086	0.9214	0.8061	0.7312	1.0
Ejecución 5	0.5923	0.7145	0.3894	0.3867	0.9002	0.8422	0.8638	0.886
Ejecución 6	0.7196	0.8184	0.4346	0.464	0.8422	0.9009	0.941	0.941
Ejecución 7	0.7028	0.818	0.3456	0.4259	0.8238	0.941	0.941	0.9222
Ejecución 8	0.6011	0.8178	0.584	0.4648	0.6593	0.8841	0.9799	0.9206
Ejecución 9	0.58	0.676	0.3454	0.3777	0.8422	0.8487	0.9799	1.0
Ejecución 10	0.5762	0.6765	0.535	0.341	0.9002	0.657	0.9799	0.9603
Monk2	0.0965 ± 0.0423	0.1144 ± 0.0368	0.2764 ± 0.0964	0.3202 ± 0.0614	0.2498 ± 0.0754	0.3051 ± 0.0552	0.3521 ± 0.0975	0.3728 ± 0.0511
Ejecución 1	0.0377	0.1126	0.3635	0.3692	0.2676	0.2402	0.2295	0.4268
Ejecución 2	0.1732	0.1065	0.1719	0.2639	0.2167	0.289	0.2657	0.3103
Ejecución 3	0.0993	0.0594	0.3815	0.3426	0.2798	0.2103	0.512	0.4329
Ejecución 4	0.1602	0.1357	0.148	0.3156	0.1969	0.3652	0.295	0.4409
Ejecución 5	0.0868	0.1946	0.3514	0.3554	0.4039	0.2636	0.2406	0.3351
Ejecución 6	0.1108	0.0829	0.3307	0.416	0.3575	0.3645	0.4249	0.3205
Ejecución 7	0.0704	0.1496	0.3298	0.2094	0.2337	0.3267	0.4227	0.3918
Ejecución 8	0.066	0.0887	0.109	0.2932	0.2149	0.2805	0.3645	0.4143
Ejecución 9	0.0485	0.1232	0.3488	0.3825	0.1576	0.3232	0.4804	0.3464
Ejecución 10	0.1125	0.0906	0.2297	0.2546	0.1694	0.3875	0.2857	0.3087
Newthyroid	0.5108 ± 0.1137	0.4953 ± 0.0325	0.4478 ± 0.0787	0.3481 ± 0.0718	0.6169 ± 0.0648	0.4817 ± 0.0596	0.6864 ± 0.0548	0.5359 ± 0.038
Ejecución 1	0.6522	0.4818	0.4093	0.2983	0.6379	0.5349	0.7125	0.4734
Ejecución 2	0.5997	0.4801	0.4575	0.3196	0.6539	0.552	0.6693	0.5161
Ejecución 3	0.5437	0.5644	0.4904	0.3532	0.6331	0.5761	0.7891	0.528
Ejecución 4	0.5011	0.5168	0.5345	0.4802	0.5659	0.4776	0.6786	0.5646
Ejecución 5	0.5588	0.474	0.4486	0.3478	0.4677	0.3785	0.6647	0.5581
Ejecución 6	0.4553	0.4488	0.3535	0.3291	0.6027	0.5068	0.6988	0.598
Ejecución 7	0.5156	0.4892	0.6019	0.369	0.7113	0.4193	0.7282	0.5007
Ejecución 8	0.4687	0.5324	0.3873	0.1905	0.6709	0.4869	0.616	0.5065
Ejecución 9	0.2177	0.4661	0.3282	0.384	0.6509	0.4596	0.5863	0.525
Ejecución 10	0.5052	0.4995	0.4666	0.4097	0.5743	0.4254	0.7202	0.5881
Page_blocks	-0.009 ± 0.0042	-0.0253 ± 0.0022	0.2165 ± 0.0164	0.2174 ± 0.042	0.3233 ± 0.0345	0.2994 ± 0.0381	0.3788 ± 0.0573	0.3215 ± 0.0663
Ejecución 1	-0.0087	-0.0265	0.2228	0.2113	0.3103	0.2906	0.3951	0.3313
Ejecución 2	-0.0073	-0.0271	0.1929	0.222	0.3486	0.2548	0.3892	0.348
Ejecución 3	-0.0206	-0.0269	0.2008	0.2858	0.3665	0.3444	0.4023	0.3131
Ejecución 4	-0.0073	-0.027	0.2314	0.2814	0.3671	0.2941	0.4474	0.2773
Ejecución 5	-0.0066	-0.0249	0.2369	0.2236	0.3088	0.2943	0.2321	0.3605
Ejecución 6	-0.0044	-0.025	0.1917	0.1629	0.3201	0.3415	0.4098	0.3784
Ejecución 7	-0.0077	-0.0231	0.2209	0.1837	0.3632	0.338	0.4343	0.1438
Ejecución 8	-0.0074	-0.0244	0.2283	0.2516	0.2896	0.2191	0.3435	0.3258
Ejecución 9	-0.01	-0.0282	0.2342	0.1808	0.2632	0.2923	0.3584	0.3566
Ejecución 10	-0.0095	-0.0205	0.2049	0.1706	0.2957	0.3249	0.3762	0.3808
Phoneme	0.2188 ± 0.0104	0.2613 ± 0.0097	0.5217 ± 0.0221	0.5813 ± 0.0411	0.7122 ± 0.0284	0.7818 ± 0.0282	0.7719 ± 0.06	0.8307 ± 0.0484
Ejecución 1	0.2122	0.2468	0.5385	0.555	0.7	0.7257	0.778	0.7306
Ejecución 2	0.2261	0.2634	0.5468	0.6274	0.725	0.8159	0.7798	0.8816
Ejecución 3	0.2067	0.2702	0.4827	0.5876	0.6441	0.7815	0.8095	0.8481
Ejecución 4	0.236	0.2462	0.5488	0.5508	0.684	0.7558	0.8179	0.7699
Ejecución 5	0.2198	0.2674	0.5117	0.5929	0.7217	0.7752	0.7941	0.8436
Ejecución 6	0.2035	0.2568	0.5256	0.575	0.7129	0.8346	0.6387	0.8794
Ejecución 7	0.2191	0.2736	0.5222	0.6527	0.7203	0.7924	0.7936	0.8743
Ejecución 8	0.2088	0.2513	0.5416	0.6098	0.7473	0.7768	0.8338	0.8459
Ejecución 9	0.2324	0.2696	0.5108	0.4999	0.7306	0.7792	0.6763	0.846
Ejecución 10	0.223	0.2678	0.4886	0.5617	0.7361	0.7812	0.7977	0.7876
Pima	0.0473 ± 0.0045	-0.0009 ± 0.0004	0.2715 ± 0.0415	0.262 ± 0.0499	0.4481 ± 0.0415	0.4705 ± 0.0632	0.5694 ± 0.0316	0.5986 ± 0.0799
Ejecución 1	0.0567	-0.0014	0.2988	0.2516	0.5211	0.5262	0.576	0.524
Ejecución 2	0.0442	-0.0008	0.2279	0.3191	0.413	0.5523	0.5523	0.701
Ejecución 3	0.0441	-0.0005	0.2748	0.2719	0.3544	0.4854	0.5485	0.5648

Continúa en la siguiente página

Tabla A.3 – continúa de la página anterior								
Nombre	0% restricciones		10% restricciones		15% restricciones		20% restricciones	
	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D
Ejecución 4	0.0517	-0.0016	0.2277	0.1877	0.4782	0.4452	0.586	0.6486
Ejecución 5	0.0461	-0.0012	0.3148	0.266	0.4434	0.5667	0.5558	0.6817
Ejecución 6	0.0518	-0.0007	0.3141	0.3097	0.4615	0.4504	0.6534	0.6291
Ejecución 7	0.0481	-0.0011	0.3328	0.2899	0.4433	0.5443	0.5792	0.5473
Ejecución 8	0.0462	-0.0001	0.2276	0.2153	0.4387	0.4875	0.5399	0.4249
Ejecución 9	0.0419	-0.0004	0.2154	0.1819	0.4723	0.3413	0.5564	0.6575
Ejecución 10	0.0426	-0.0008	0.2813	0.3271	0.4549	0.4153	0.5464	0.6073
Saheart	-0.0035 ± 0.0165	-0.0079 ± 0.0081	0.1341 ± 0.0251	0.1627 ± 0.0346	0.1983 ± 0.0192	0.2174 ± 0.0349	0.305 ± 0.0334	0.3606 ± 0.0382
Ejecución 1	0.0125	-0.0119	0.1553	0.1318	0.2338	0.2464	0.3169	0.3513
Ejecución 2	0.0179	-0.0095	0.0721	0.2114	0.1822	0.2505	0.2928	0.3777
Ejecución 3	0.0015	0.0087	0.1421	0.1934	0.1838	0.2144	0.2831	0.4364
Ejecución 4	0.003	0.0008	0.1197	0.1326	0.2198	0.2336	0.3253	0.3373
Ejecución 5	-0.0303	-0.0096	0.1401	0.2222	0.1744	0.2377	0.3069	0.3655
Ejecución 6	-0.0264	-0.0146	0.1388	0.1315	0.1737	0.1569	0.3093	0.3505
Ejecución 7	0.0074	-0.0186	0.1389	0.1861	0.2151	0.225	0.2417	0.4158
Ejecución 8	-0.0011	-0.0112	0.1202	0.1352	0.2017	0.2543	0.378	0.3089
Ejecución 9	-0.0256	-0.0138	0.1432	0.1456	0.2008	0.152	0.2818	0.3181
Ejecución 10	0.0058	0.0005	0.1712	0.1371	0.1976	0.2035	0.3142	0.345
Satimage	0.3654 ± 0.0036	0.4634 ± 0.013	0.744 ± 0.0082	0.8213 ± 0.0205	0.8204 ± 0.0143	0.8986 ± 0.0117	0.8444 ± 0.0422	0.9205 ± 0.026
Ejecución 1	0.367	0.4846	0.7374	0.8571	0.8197	0.9032	0.8655	0.8678
Ejecución 2	0.3651	0.4822	0.7573	0.8495	0.8198	0.9013	0.8034	0.9308
Ejecución 3	0.365	0.4618	0.7412	0.8211	0.8186	0.9171	0.8287	0.9447
Ejecución 4	0.3619	0.4713	0.7349	0.8153	0.8296	0.9188	0.8682	0.949
Ejecución 5	0.3679	0.458	0.7433	0.8267	0.8194	0.8932	0.8748	0.9395
Ejecución 6	0.371	0.4642	0.7303	0.7824	0.8137	0.8955	0.7506	0.9244
Ejecución 7	0.3571	0.4655	0.7536	0.811	0.8187	0.9014	0.811	0.9452
Ejecución 8	0.3678	0.4518	0.7428	0.8278	0.7945	0.8858	0.8878	0.8941
Ejecución 9	0.3648	0.4396	0.7476	0.8017	0.8145	0.8843	0.8831	0.9168
Ejecución 10	0.3668	0.455	0.7518	0.8201	0.8551	0.8854	0.8709	0.8925
Segment	0.2303 ± 0.0319	0.2606 ± 0.0102	0.2875 ± 0.0263	0.2911 ± 0.0271	0.3734 ± 0.0391	0.3958 ± 0.0625	0.639 ± 0.0508	0.6725 ± 0.0893
Ejecución 1	0.2634	0.2616	0.2713	0.3185	0.3711	0.5012	0.6033	0.7537
Ejecución 2	0.1962	0.2448	0.2675	0.2741	0.3974	0.3121	0.6418	0.7687
Ejecución 3	0.1968	0.2745	0.2242	0.2542	0.3923	0.4167	0.5997	0.5447
Ejecución 4	0.2107	0.2708	0.3009	0.3056	0.3955	0.4612	0.6144	0.6784
Ejecución 5	0.1896	0.2712	0.3221	0.2515	0.4459	0.4118	0.702	0.7431
Ejecución 6	0.2636	0.2522	0.2933	0.3278	0.3096	0.3514	0.6137	0.5284
Ejecución 7	0.2627	0.2654	0.2914	0.2943	0.3517	0.3922	0.694	0.7871
Ejecución 8	0.2021	0.2619	0.3139	0.2611	0.3934	0.3332	0.7052	0.6107
Ejecución 9	0.2675	0.2436	0.2946	0.3174	0.314	0.3182	0.5428	0.7008
Ejecución 10	0.2502	0.26	0.2955	0.3065	0.3629	0.4596	0.673	0.6092
Sonar	0.0002 ± 0.0068	-0.0045 ± 0.0003	0.0233 ± 0.0218	0.0255 ± 0.0252	0.0381 ± 0.0284	0.0283 ± 0.0249	0.0646 ± 0.0393	0.1055 ± 0.0384
Ejecución 1	0.0065	-0.0039	0.0134	0.0134	0.0134	0.0253	0.068	0.0903
Ejecución 2	-0.0046	-0.0039	-0.0048	0.058	0.0487	0.0443	0.109	0.109
Ejecución 3	-0.0024	-0.0046	0.0487	-0.004	0.058	-0.0047	0.0681	0.109
Ejecución 4	-0.0039	-0.0047	0.0401	0.0287	0.0733	0.0443	-0.0003	0.0064
Ejecución 5	-0.0032	-0.0046	0.0287	-0.0047	0.0443	-0.0044	0.0532	0.1293
Ejecución 6	0.0162	-0.0044	0.019	0.0487	0.0904	-0.0044	0.1293	0.109
Ejecución 7	-0.0047	-0.0047	-0.0047	0.0629	0.019	0.0324	0.0788	0.0845
Ejecución 8	-0.0044	-0.0046	0.0532	0.0109	-0.0044	0.0487	-0.0025	0.1293
Ejecución 9	0.0065	-0.0044	0.0443	0.0443	0.0161	0.0733	0.0789	0.1365
Ejecución 10	-0.0043	-0.0047	-0.0045	-0.0033	0.022	0.0288	0.063	0.1513
Soybean	0.5862 ± 0.0146	0.5484 ± 0.0442	0.5588 ± 0.0866	0.5923 ± 0.0881	0.7601 ± 0.2072	0.7496 ± 0.1661	0.4928 ± 0.1424	0.4514 ± 0.0567
Ejecución 1	0.5519	0.4768	0.6097	0.529	0.8754	1.0	0.2815	0.4263
Ejecución 2	0.5855	0.4917	0.6077	0.517	0.517	0.8201	0.2151	0.4529
Ejecución 3	0.5855	0.5678	0.5611	0.5741	0.8754	0.5535	0.5442	0.447
Ejecución 4	0.5855	0.5814	0.5611	0.5951	0.5135	0.5328	0.6011	0.4337
Ejecución 5	0.5855	0.5634	0.3076	0.4807	0.4555	0.6676	0.3739	0.4167
Ejecución 6	0.5855	0.5814	0.6097	0.6527	0.63	1.0	0.5906	0.464
Ejecución 7	0.5855	0.4768	0.6097	0.539	1.0	0.5719	0.5202	0.5591
Ejecución 8	0.5855	0.5814	0.5616	0.8201	0.7347	0.6851	0.5678	0.3703
Ejecución 9	0.6153	0.5814	0.6097	0.5485	1.0	0.8758	0.563	0.5455
Ejecución 10	0.5963	0.5814	0.5963	0.5762	1.0	0.7893	0.6708	0.3985
Spambase	0.0606 ± 0.0035	0.0732 ± 0.0035	0.3822 ± 0.0289	0.4478 ± 0.035	0.5949 ± 0.0269	0.5978 ± 0.0327	0.5642 ± 0.1409	0.663 ± 0.1058
Ejecución 1	0.058	0.0746	0.3843	0.4344	0.6185	0.6328	0.6361	0.6889
Ejecución 2	0.0624	0.0697	0.4134	0.4715	0.6207	0.5358	0.3122	0.7381
Ejecución 3	0.0646	0.0713	0.361	0.423	0.5715	0.5842	0.444	0.6867
Ejecución 4	0.059	0.0668	0.3412	0.4132	0.5806	0.616	0.7082	0.6964
Ejecución 5	0.0573	0.0731	0.3861	0.5269	0.6531	0.5869	0.5417	0.7355
Ejecución 6	0.0572	0.0734	0.3664	0.4377	0.561	0.6535	0.3371	0.3526
Ejecución 7	0.0661	0.0766	0.4068	0.4268	0.5819	0.5759	0.6624	0.6646
Ejecución 8	0.0572	0.0774	0.4306	0.4303	0.5986	0.573	0.6387	0.6842
Ejecución 9	0.0656	0.071	0.3384	0.4917	0.5718	0.6258	0.6893	0.7011
Ejecución 10	0.0583	0.0783	0.3932	0.4225	0.5914	0.5942	0.6723	0.6815
Spectfheart	-0.0997 ± 0.0022	-0.105 ± 0.001	0.1476 ± 0.0498	0.1131 ± 0.0511	0.2418 ± 0.1063	0.2361 ± 0.1027	0.258 ± 0.123	0.2677 ± 0.1195
Ejecución 1	-0.0996	-0.1048	0.1646	0.1577	0.2333	0.2507	0.2401	0.4143
Ejecución 2	-0.0996	-0.1066	0.0781	0.0702	0.1731	0.1031	0.3773	0.3496
Ejecución 3	-0.0984	-0.1034	0.1471	0.1378	0.4104	0.3547	0.5165	0.3198
Ejecución 4	-0.1017	-0.1041	0.1806	0.1596	0.2942	0.1595	0.2231	0.2241
Ejecución 5	-0.0984	-0.1053	0.0792	0.1874	0.3037	0.094	0.008	0.082
Ejecución 6	-0.0957	-0.1041	0.2005	0.0459	0.2628	0.3125	0.3038	0.087
Ejecución 7	-0.1041	-0.1053	0.2188	0.0178	-0.0157	0.4007	0.2186	0.4021
Ejecución 8	-0.0984	-0.1048	0.1446	0.1246	0.2551	0.2767	0.2325	0.1519
Ejecución 9	-0.1017	-0.1067	0.1834	0.1052	0.1957	0.2783	0.2186	0.3748
Ejecución 10	-0.0996	-0.1053	0.0789	0.1251	0.3057	0.1303	0.2411	0.2707
Tae	0.0057 ± 0.0041	0.0203 ± 0.0172	0.017 ± 0.0162	0.0332 ± 0.0368	0.0245 ± 0.0289	0.0176 ± 0.0097	0.0298 ± 0.0233	0.0307 ± 0.0257
Ejecución 1	0.0018	0.0372	0.0039	0.0174	0.0065	0.0241	0.0176	0.0166
Ejecución 2	0.0068	0.0399	0.0028	0.0439	0.0363	0.0243	0.0469	-0.0082
Ejecución 3	0.0069	-0.0046	-0.0024	0.1109	0.0066	0.0133	0.0379	0.0604
Ejecución 4	0.0052	-0.0052	0.0499	0.0554	0.0297	0.0385	0.065	0.0058
Ejecución 5	0.0089	0.0369	0.0357	-0.0056	-0.0007	0.0246	0.0682	0.0369
Ejecución 6	0.0027	0.0327	0.0269	0.0406	0.0049	0.006	0.0275	0.0428

Continúa en la siguiente página

A. Resultados experimentales completos.

Tabla A.3 – continúa de la página anterior								
Nombre	0 % restricciones		10 % restricciones		15 % restricciones		20 % restricciones	
	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D
Ejecución 7	0.0086	-0.0019	0.0196	0.0139	0.0966	0.0074	0.0044	0.0206
Ejecución 8	0.0144	0.03	0.013	0.0703	0.0108	0.0131	0.0053	0.0046
Ejecución 9	0.0017	0.019	-0.0008	-0.008	0.051	0.0167	0.0255	0.0749
Ejecución 10	-0.0001	0.019	0.0218	-0.007	0.0037	0.0077	-0.0008	0.053
Thyroid	-0.0251 ± 0.0118	-0.0077 ± 0.001	0.299 ± 0.2172	0.774 ± 0.0351	0.4168 ± 0.2183	0.9268 ± 0.0491	0.3631 ± 0.2177	0.9128 ± 0.116
Ejecución 1	-0.0121	-0.007	0.1934	0.7649	0.6005	0.9536	0.7352	0.989
Ejecución 2	-0.0329	-0.0096	0.0781	0.7248	0.0929	0.886	0.122	1.0
Ejecución 3	-0.0328	-0.0084	0.0002	0.7552	0.5728	0.8516	0.1436	0.9694
Ejecución 4	-0.0321	-0.007	0.0819	0.7395	0.2267	0.9573	0.2393	0.9702
Ejecución 5	-0.0321	-0.0084	0.5057	0.787	0.7046	0.9575	0.2704	0.861
Ejecución 6	-0.0328	-0.0078	0.4232	0.7291	0.5149	0.9382	0.5898	0.9815
Ejecución 7	-0.0012	-0.0072	0.0922	0.8222	0.2174	0.9686	0.2585	0.9696
Ejecución 8	-0.0339	-0.0086	0.5096	0.7825	0.5853	0.9322	0.3679	0.7046
Ejecución 9	-0.0321	-0.0067	0.5287	0.8176	0.5504	0.8346	0.2002	0.6824
Ejecución 10	-0.0091	-0.006	0.5772	0.8167	0.1024	0.9886	0.7042	1.0
Titanic	0.0512 ± 0.0814	0.1484 ± 0.0484	0.4293 ± 0.0343	0.4654 ± 0.0284	0.555 ± 0.047	0.563 ± 0.0408	0.677 ± 0.0387	0.7197 ± 0.0513
Ejecución 1	-0.0035	0.1095	0.4567	0.4327	0.587	0.537	0.6211	0.7222
Ejecución 2	0.2127	0.2159	0.3714	0.4532	0.5502	0.565	0.633	0.7122
Ejecución 3	0.1856	0.2125	0.3935	0.4327	0.5698	0.5132	0.6737	0.6357
Ejecución 4	-0.0035	0.1095	0.4246	0.4958	0.5123	0.5194	0.6687	0.7516
Ejecución 5	-0.0035	0.1095	0.4299	0.4545	0.6454	0.6074	0.7528	0.6376
Ejecución 6	0.0248	0.2159	0.4118	0.4795	0.4802	0.6324	0.7272	0.7554
Ejecución 7	-0.0035	0.1095	0.4276	0.5053	0.5364	0.6194	0.6407	0.811
Ejecución 8	0.1095	0.1095	0.5043	0.4366	0.5745	0.5652	0.6929	0.7171
Ejecución 9	-0.0035	0.1822	0.4237	0.4537	0.5006	0.5291	0.6802	0.7567
Ejecución 10	-0.0035	0.1095	0.4494	0.5096	0.594	0.5417	0.6801	0.698
Vehicle	0.0894 ± 0.0053	0.0695 ± 0.0036	0.1215 ± 0.0245	0.1127 ± 0.0303	0.3524 ± 0.0339	0.3733 ± 0.0665	0.5551 ± 0.0516	0.6141 ± 0.0264
Ejecución 1	0.08	0.0699	0.1142	0.139	0.3366	0.4146	0.5487	0.6186
Ejecución 2	0.0882	0.0771	0.0916	0.139	0.4208	0.2943	0.5107	0.639
Ejecución 3	0.0888	0.0713	0.1388	0.1113	0.3563	0.3059	0.579	0.6088
Ejecución 4	0.088	0.0714	0.1707	0.1168	0.3585	0.3375	0.4753	0.6113
Ejecución 5	0.0925	0.0708	0.0881	0.0651	0.3126	0.4742	0.5881	0.5878
Ejecución 6	0.0803	0.0701	0.1159	0.1322	0.3044	0.3317	0.6544	0.6637
Ejecución 7	0.0937	0.065	0.1434	0.1414	0.3894	0.4077	0.5439	0.6412
Ejecución 8	0.0952	0.0648	0.1339	0.0676	0.3709	0.4522	0.4857	0.5689
Ejecución 9	0.091	0.0693	0.1199	0.0738	0.3218	0.4299	0.5717	0.6056
Ejecución 10	0.096	0.065	0.0985	0.1406	0.3531	0.2847	0.5937	0.5965
Wdbc	0.2824 ± 0.0212	0.3576 ± 0.0122	0.7698 ± 0.0261	0.7936 ± 0.0152	0.7507 ± 0.0288	0.7682 ± 0.0307	0.8272 ± 0.0245	0.8304 ± 0.0247
Ejecución 1	0.2537	0.3631	0.7364	0.7797	0.7672	0.761	0.8311	0.8373
Ejecución 2	0.3089	0.3538	0.8052	0.7985	0.7863	0.7987	0.8052	0.8308
Ejecución 3	0.2872	0.3447	0.7436	0.8049	0.7135	0.7989	0.8373	0.7736
Ejecución 4	0.2787	0.3724	0.7928	0.7798	0.7303	0.7799	0.7926	0.8634
Ejecución 5	0.2456	0.3447	0.7804	0.7922	0.7249	0.7797	0.8766	0.8308
Ejecución 6	0.3045	0.3771	0.7676	0.7925	0.743	0.7005	0.8371	0.7994
Ejecución 7	0.2703	0.3401	0.7306	0.7795	0.7074	0.7797	0.8051	0.8308
Ejecución 8	0.3045	0.3492	0.7804	0.7735	0.7742	0.7244	0.8182	0.8438
Ejecución 9	0.3002	0.3681	0.7557	0.8179	0.7741	0.7924	0.857	0.8503
Ejecución 10	0.2703	0.3631	0.8052	0.8178	0.7859	0.7672	0.812	0.8438
Wine	0.2147 ± 0.0112	0.332 ± 0.0134	0.311 ± 0.0718	0.2637 ± 0.0916	0.4717 ± 0.0636	0.488 ± 0.0545	0.6731 ± 0.0597	0.6814 ± 0.0512
Ejecución 1	0.2171	0.3036	0.2807	0.1817	0.5687	0.5125	0.6964	0.6487
Ejecución 2	0.2294	0.3512	0.4328	0.1508	0.5866	0.4714	0.6588	0.7421
Ejecución 3	0.2156	0.3334	0.3534	0.3718	0.5053	0.4983	0.6142	0.6602
Ejecución 4	0.2085	0.3395	0.2178	0.431	0.4037	0.4392	0.7566	0.645
Ejecución 5	0.2047	0.339	0.4301	0.1644	0.4392	0.3977	0.7063	0.6927
Ejecución 6	0.2261	0.3263	0.2831	0.355	0.4386	0.4526	0.5991	0.7229
Ejecución 7	0.2206	0.3434	0.3286	0.1914	0.5074	0.5279	0.6062	0.7075
Ejecución 8	0.193	0.3416	0.24	0.2923	0.4079	0.6047	0.6144	0.6938
Ejecución 9	0.2047	0.3193	0.3034	0.2564	0.4066	0.5169	0.7168	0.7378
Ejecución 10	0.2272	0.3227	0.2404	0.2422	0.4534	0.4584	0.7625	0.5639
Zoo	0.6575 ± 0.035	0.6476 ± 0.0151	0.3783 ± 0.0863	0.355 ± 0.0394	0.4694 ± 0.1216	0.4698 ± 0.0748	0.6145 ± 0.0931	0.6477 ± 0.1238
Ejecución 1	0.6654	0.6491	0.6004	0.3891	0.4211	0.4609	0.635	0.6111
Ejecución 2	0.6154	0.6415	0.3633	0.3732	0.3924	0.6508	0.5164	0.7529
Ejecución 3	0.6875	0.6631	0.2777	0.3484	0.6947	0.4798	0.6981	0.7286
Ejecución 4	0.6719	0.6545	0.3536	0.4302	0.5374	0.3665	0.589	0.8408
Ejecución 5	0.6596	0.6564	0.3452	0.3405	0.3694	0.4244	0.8099	0.5724
Ejecución 6	0.6681	0.634	0.3101	0.2722	0.6251	0.4585	0.6449	0.584
Ejecución 7	0.6488	0.6096	0.4062	0.3503	0.3652	0.5403	0.494	0.4642
Ejecución 8	0.5756	0.6507	0.3052	0.3464	0.5793	0.4178	0.6715	0.4589
Ejecución 9	0.697	0.6587	0.3978	0.326	0.3371	0.4205	0.5126	0.7757
Ejecución 10	0.6863	0.6587	0.4232	0.3739	0.3722	0.4788	0.5737	0.6882
Media	0.1707 ± 0.2108	0.1907 ± 0.2257	0.321 ± 0.1976	0.3396 ± 0.2309	0.4538 ± 0.2228	0.4762 ± 0.2414	0.5107 ± 0.2272	0.5476 ± 0.2424

Tabla A.4.: Resultados de las ejecuciones del SCCSOM-E.

Nombre	0% restricciones		10% restricciones		15% restricciones		20% restricciones	
	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D
Appendicitis	-0.0958 ± 0.0	-0.0958 ± 0.0	0.1274 ± 0.3046	0.8317 ± 0.0	0.3129 ± 0.1132	0.3506 ± 0.0	0.8004 ± 0.0	0.8375 ± 0.0
Ejecución 1	-0.0958	-0.0958	-0.0602	0.8317	0.3506	0.3506	0.8004	0.8375
Ejecución 2	-0.0958	-0.0958	0.8317	0.8317	-0.0266	0.3506	0.8004	0.8375
Ejecución 3	-0.0958	-0.0958	0.4318	0.8317	0.3506	0.3506	0.8004	0.8375
Ejecución 4	-0.0958	-0.0958	-0.0602	0.8317	0.3506	0.3506	0.8004	0.8375
Ejecución 5	-0.0958	-0.0958	-0.0602	0.8317	0.3506	0.3506	0.8004	0.8375
Ejecución 6	-0.0958	-0.0958	-0.0602	0.8317	0.3506	0.3506	0.8004	0.8375
Ejecución 7	-0.0958	-0.0958	-0.0602	0.8317	0.3506	0.3506	0.8004	0.8375
Ejecución 8	-0.0958	-0.0958	-0.0602	0.8317	0.3506	0.3506	0.8004	0.8375
Ejecución 9	-0.0958	-0.0958	-0.0602	0.8317	0.3506	0.3506	0.8004	0.8375
Ejecución 10	-0.0958	-0.0958	0.4318	0.8317	0.3506	0.3506	0.8004	0.8375
Balance	0.1189 ± 0.0	0.1122 ± 0.0	0.856 ± 0.001	0.8795 ± 0.1305	0.8635 ± 0.0005	1.0 ± 0.0	0.8566 ± 0.0022	1.0 ± 0.0
Ejecución 1	0.1189	0.1122	0.8564	0.986	0.8637	1.0	0.8577	1.0
Ejecución 2	0.1189	0.1122	0.8564	0.986	0.8637	1.0	0.8522	1.0
Ejecución 3	0.1189	0.1122	0.8564	0.986	0.8637	1.0	0.8577	1.0
Ejecución 4	0.1189	0.1122	0.853	0.7197	0.8626	1.0	0.8577	1.0
Ejecución 5	0.1189	0.1122	0.8564	0.7197	0.8637	1.0	0.8577	1.0
Ejecución 6	0.1189	0.1122	0.8564	0.986	0.8637	1.0	0.8577	1.0
Ejecución 7	0.1189	0.1122	0.8564	0.7197	0.8637	1.0	0.8522	1.0
Ejecución 8	0.1189	0.1122	0.8562	0.7197	0.8626	1.0	0.8577	1.0
Ejecución 9	0.1189	0.1122	0.8564	0.986	0.8637	1.0	0.8577	1.0
Ejecución 10	0.1189	0.1122	0.8564	0.986	0.8637	1.0	0.8577	1.0
Banana	-0.0006 ± 0.0	-0.0006 ± 0.0	0.8978 ± 0.2992	0.9 ± 0.2999	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	-0.0006	-0.0006	0.9975	1.0	1.0	1.0	1.0	1.0
Ejecución 2	-0.0006	-0.0006	0.9975	1.0	1.0	1.0	1.0	1.0
Ejecución 3	-0.0006	-0.0006	0.0003	1.0	1.0	1.0	1.0	1.0
Ejecución 4	-0.0006	-0.0006	0.9975	1.0	1.0	1.0	1.0	1.0
Ejecución 5	-0.0006	-0.0006	0.9975	1.0	1.0	1.0	1.0	1.0
Ejecución 6	-0.0006	-0.0006	0.9975	1.0	1.0	1.0	1.0	1.0
Ejecución 7	-0.0006	-0.0006	0.9975	1.0	1.0	1.0	1.0	1.0
Ejecución 8	-0.0006	-0.0006	0.9975	1.0	1.0	1.0	1.0	1.0
Ejecución 9	-0.0006	-0.0006	0.9975	1.0	1.0	1.0	1.0	1.0
Ejecución 10	-0.0006	-0.0006	0.9975	1.0	1.0	1.0	1.0	1.0
Breast_cancer	0.009 ± 0.0	0.009 ± 0.0	0.6979 ± 0.215	0.3926 ± 0.0924	0.9916 ± 0.0103	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	0.009	0.009	0.5159	0.3172	0.9719	1.0	1.0	1.0
Ejecución 2	0.009	0.009	0.5159	0.3172	0.9719	1.0	1.0	1.0
Ejecución 3	0.009	0.009	0.9036	0.3172	1.0	1.0	1.0	1.0
Ejecución 4	0.009	0.009	0.9036	0.5058	0.9929	1.0	1.0	1.0
Ejecución 5	0.009	0.009	0.8638	0.3172	1.0	1.0	1.0	1.0
Ejecución 6	0.009	0.009	0.5159	0.5058	0.9929	1.0	1.0	1.0
Ejecución 7	0.009	0.009	0.2731	0.5058	0.9929	1.0	1.0	1.0
Ejecución 8	0.009	0.009	0.8638	0.3172	1.0	1.0	1.0	1.0
Ejecución 9	0.009	0.009	0.7199	0.5058	1.0	1.0	1.0	1.0
Ejecución 10	0.009	0.009	0.9036	0.3172	0.9929	1.0	1.0	1.0
Contraceptive	0.1165 ± 0.0	0.0811 ± 0.0	0.7097 ± 0.2739	0.9964 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	0.1165	0.0811	0.6844	0.9964	1.0	1.0	1.0	1.0
Ejecución 2	0.1165	0.0811	0.6864	0.9964	1.0	1.0	1.0	1.0
Ejecución 3	0.1165	0.0811	0.0006	0.9964	1.0	1.0	1.0	1.0
Ejecución 4	0.1165	0.0811	0.9941	0.9964	1.0	1.0	1.0	1.0
Ejecución 5	0.1165	0.0811	0.6854	0.9964	1.0	1.0	1.0	1.0
Ejecución 6	0.1165	0.0811	0.9958	0.9964	1.0	1.0	1.0	1.0
Ejecución 7	0.1165	0.0811	0.9941	0.9964	1.0	1.0	1.0	1.0
Ejecución 8	0.1165	0.0811	0.6844	0.9964	1.0	1.0	1.0	1.0
Ejecución 9	0.1165	0.0811	0.6854	0.9964	1.0	1.0	1.0	1.0
Ejecución 10	0.1165	0.0811	0.6864	0.9964	1.0	1.0	1.0	1.0
Ecoli	0.7944 ± 0.0	0.7497 ± 0.0	0.7981 ± 0.0208	0.8361 ± 0.0003	0.8953 ± 0.01	0.8903 ± 0.0006	0.9549 ± 0.0081	0.9432 ± 0.0
Ejecución 1	0.7944	0.7497	0.8248	0.8359	0.9098	0.8901	0.9492	0.9432
Ejecución 2	0.7944	0.7497	0.8248	0.8367	0.9098	0.8901	0.9492	0.9432
Ejecución 3	0.7944	0.7497	0.8384	0.8359	0.9098	0.8901	0.9492	0.9432
Ejecución 4	0.7944	0.7497	0.7846	0.8359	0.8985	0.8901	0.9361	0.9432
Ejecución 5	0.7944	0.7497	0.7846	0.8367	0.8875	0.8901	0.9608	0.9432
Ejecución 6	0.7944	0.7497	0.7846	0.8359	0.8875	0.8901	0.9608	0.9432
Ejecución 7	0.7944	0.7497	0.7846	0.8359	0.8875	0.8901	0.9608	0.9432
Ejecución 8	0.7944	0.7497	0.7846	0.8359	0.8875	0.8901	0.9608	0.9432
Ejecución 9	0.7944	0.7497	0.7846	0.8359	0.8875	0.8901	0.9608	0.9432
Ejecución 10	0.7944	0.7497	0.7846	0.8359	0.8875	0.8919	0.9608	0.9432
Glass	0.543 ± 0.0	0.5153 ± 0.0	0.5632 ± 0.1203	0.5692 ± 0.026	0.7428 ± 0.0176	0.7247 ± 0.0114	0.8723 ± 0.0207	0.8647 ± 0.0158
Ejecución 1	0.543	0.5153	0.6305	0.592	0.7499	0.7172	0.8148	0.8544
Ejecución 2	0.543	0.5153	0.3994	0.5648	0.7524	0.7172	0.8834	0.8544
Ejecución 3	0.543	0.5153	0.3994	0.541	0.7527	0.7421	0.8834	0.8888
Ejecución 4	0.543	0.5153	0.6733	0.5648	0.7628	0.7172	0.8834	0.8544
Ejecución 5	0.543	0.5153	0.5524	0.592	0.7251	0.7172	0.8751	0.8544
Ejecución 6	0.543	0.5153	0.7095	0.541	0.7527	0.7172	0.8834	0.8544
Ejecución 7	0.543	0.5153	0.5216	0.6262	0.7213	0.7421	0.8751	0.8888
Ejecución 8	0.543	0.5153	0.6733	0.541	0.7524	0.7421	0.8834	0.8888
Ejecución 9	0.543	0.5153	0.6733	0.5648	0.7527	0.7172	0.8834	0.8544
Ejecución 10	0.543	0.5153	0.3994	0.5648	0.7053	0.7172	0.8574	0.8544
Haberman	-0.0159 ± 0.0	-0.0159 ± 0.0	0.3385 ± 0.1143	0.0268 ± 0.0217	0.9674 ± 0.0498	1.0 ± 0.0	0.8797 ± 0.2452	1.0 ± 0.0
Ejecución 1	-0.0159	-0.0159	0.3119	0.0003	0.8914	1.0	1.0	1.0
Ejecución 2	-0.0159	-0.0159	0.405	0.0445	1.0	1.0	0.878	1.0
Ejecución 3	-0.0159	-0.0159	0.3119	0.0445	0.8914	1.0	1.0	1.0
Ejecución 4	-0.0159	-0.0159	0.405	0.0445	1.0	1.0	0.878	1.0
Ejecución 5	-0.0159	-0.0159	0.3119	0.0003	1.0	1.0	1.0	1.0
Ejecución 6	-0.0159	-0.0159	0.405	0.0003	1.0	1.0	0.878	1.0
Ejecución 7	-0.0159	-0.0159	0.4441	0.0003	1.0	1.0	1.0	1.0
Ejecución 8	-0.0159	-0.0159	0.0348	0.0445	1.0	1.0	1.0	1.0
Ejecución 9	-0.0159	-0.0159	0.4441	0.0445	1.0	1.0	0.1625	1.0
Ejecución 10	-0.0159	-0.0159	0.3119	0.0445	0.8914	1.0	1.0	1.0
Hayes_roth	-0.0053 ± 0.0	-0.0087 ± 0.0	0.0305 ± 0.0064	0.0525 ± 0.0249	0.0019 ± 0.0059	0.0013 ± 0.0045	0.303 ± 0.1381	0.2359 ± 0.0

Continúa en la siguiente página

A. Resultados experimentales completos.

Tabla A.4 – continúa de la página anterior

Nombre	0 % restricciones		10 % restricciones		15 % restricciones		20 % restricciones	
	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D
Ejecución 1	-0.0053	-0.0087	0.0331	0.0026	-0.004	0.0912	0.2359	0.2359
Ejecución 2	-0.0053	-0.0087	0.0331	0.0649	0.0012	-0.0002	0.2124	0.2359
Ejecución 3	-0.0053	-0.0087	0.0131	0.0026	-0.0005	-0.0002	0.5169	0.2359
Ejecución 4	-0.0053	-0.0087	0.0331	0.0649	-0.004	-0.0002	0.4307	0.2359
Ejecución 5	-0.0053	-0.0087	0.0279	0.0649	0.0059	-0.0002	0.3481	0.2359
Ejecución 6	-0.0053	-0.0087	0.0331	0.0649	0.0012	-0.0002	0.2124	0.2359
Ejecución 7	-0.0053	-0.0087	0.0279	0.0649	0.0161	-0.0002	0.3481	0.2359
Ejecución 8	-0.0053	-0.0087	0.0331	0.0649	-0.004	0.0148	0.4307	0.2359
Ejecución 9	-0.0053	-0.0087	0.0378	0.0649	0.0059	-0.0002	0.3481	0.2359
Ejecución 10	-0.0053	-0.0087	0.0331	0.0649	0.0012	-0.0002	0.0912	0.2359
Heart	0.0716 ± 0.0	0.0716 ± 0.0	0.0955 ± 0.0687	0.2881 ± 0.0145	0.6933 ± 0.2908	0.9309 ± 0.0609	0.9077 ± 0.2768	1.0 ± 0.0
Ejecución 1	0.0716	0.0716	0.0264	0.266	0.3212	0.8295	1.0	1.0
Ejecución 2	0.0716	0.0716	0.123	0.266	0.9705	0.8568	1.0	1.0
Ejecución 3	0.0716	0.0716	0.0054	0.266	0.9705	0.8295	1.0	1.0
Ejecución 4	0.0716	0.0716	0.0055	0.2976	0.6627	0.9705	0.0772	1.0
Ejecución 5	0.0716	0.0716	0.123	0.2976	0.9705	0.9705	1.0	1.0
Ejecución 6	0.0716	0.0716	0.1028	0.2976	0.3656	0.9705	1.0	1.0
Ejecución 7	0.0716	0.0716	0.1231	0.2976	0.3656	0.9705	1.0	1.0
Ejecución 8	0.0716	0.0716	0.1028	0.2976	0.3656	0.9705	1.0	1.0
Ejecución 9	0.0716	0.0716	0.0935	0.2976	0.9705	0.9705	1.0	1.0
Ejecución 10	0.0716	0.0716	0.2498	0.2976	0.9705	0.9705	1.0	1.0
Ionosphere	-0.0059 ± 0.0	-0.0059 ± 0.0	0.0375 ± 0.0511	0.2071 ± 0.1301	0.9673 ± 0.0379	0.9863 ± 0.0112	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	-0.0059	-0.0059	0.0157	0.1079	0.9771	0.9771	1.0	1.0
Ejecución 2	-0.0059	-0.0059	0.0157	0.1079	0.9771	0.9771	1.0	1.0
Ejecución 3	-0.0059	-0.0059	0.1412	0.1802	1.0	1.0	1.0	1.0
Ejecución 4	-0.0059	-0.0059	0.0157	0.1079	0.9771	0.9771	1.0	1.0
Ejecución 5	-0.0059	-0.0059	0.0095	0.1802	0.8555	1.0	1.0	1.0
Ejecución 6	-0.0059	-0.0059	0.0109	0.4592	0.9771	0.9771	1.0	1.0
Ejecución 7	-0.0059	-0.0059	-0.0023	0.1802	0.9771	1.0	1.0	1.0
Ejecución 8	-0.0059	-0.0059	0.0157	0.4592	0.9771	0.9771	1.0	1.0
Ejecución 9	-0.0059	-0.0059	0.0157	0.1079	0.9771	0.9771	1.0	1.0
Ejecución 10	-0.0059	-0.0059	0.137	0.1802	0.9771	1.0	1.0	1.0
Iris	0.6562 ± 0.0	0.6295 ± 0.0	0.4481 ± 0.0968	0.5794 ± 0.0	0.6473 ± 0.1412	0.8857 ± 0.0	0.1927 ± 0.2801	0.9603 ± 0.0
Ejecución 1	0.6562	0.6295	0.5632	0.5794	0.5745	0.8857	0.5371	0.9603
Ejecución 2	0.6562	0.6295	0.3518	0.5794	0.5239	0.8857	0.0187	0.9603
Ejecución 3	0.6562	0.6295	0.5718	0.5794	0.7011	0.8857	0.0169	0.9603
Ejecución 4	0.6562	0.6295	0.5632	0.5794	0.5073	0.8857	0.7599	0.9603
Ejecución 5	0.6562	0.6295	0.3518	0.5794	0.5239	0.8857	0.0187	0.9603
Ejecución 6	0.6562	0.6295	0.3882	0.5794	0.8478	0.8857	0.0067	0.9603
Ejecución 7	0.6562	0.6295	0.3518	0.5794	0.8478	0.8857	0.0067	0.9603
Ejecución 8	0.6562	0.6295	0.3882	0.5794	0.8478	0.8857	0.0067	0.9603
Ejecución 9	0.6562	0.6295	0.3882	0.5794	0.5239	0.8857	0.0187	0.9603
Ejecución 10	0.6562	0.6295	0.5632	0.5794	0.5745	0.8857	0.5371	0.9603
Monk2	0.0178 ± 0.0	0.0191 ± 0.0	0.8039 ± 0.0731	0.7602 ± 0.1022	0.9781 ± 0.0044	0.9814 ± 0.0015	0.9799 ± 0.0029	0.9817 ± 0.0003
Ejecución 1	0.0178	0.0191	0.8776	0.8837	0.9816	0.9827	0.9811	0.9811
Ejecución 2	0.0178	0.0191	0.8702	0.8837	0.9778	0.9811	0.9816	0.9816
Ejecución 3	0.0178	0.0191	0.8233	0.8715	0.9767	0.9827	0.9821	0.9816
Ejecución 4	0.0178	0.0191	0.7129	0.6225	0.9811	0.9816	0.9772	0.9816
Ejecución 5	0.0178	0.0191	0.6637	0.7243	0.9687	0.9821	0.9806	0.9816
Ejecución 6	0.0178	0.0191	0.8323	0.6734	0.9821	0.9772	0.9816	0.9821
Ejecución 7	0.0178	0.0191	0.8587	0.7243	0.9816	0.9816	0.9784	0.9821
Ejecución 8	0.0178	0.0191	0.7129	0.8715	0.9724	0.9816	0.9724	0.9821
Ejecución 9	0.0178	0.0191	0.8497	0.7243	0.9767	0.9816	0.9816	0.9821
Ejecución 10	0.0178	0.0191	0.838	0.6225	0.9821	0.9816	0.9821	0.9816
Newthyroid	0.9192 ± 0.0	0.8928 ± 0.0	0.9243 ± 0.0084	0.9214 ± 0.0	0.7852 ± 0.1873	0.9079 ± 0.0	0.9541 ± 0.0197	0.9602 ± 0.0
Ejecución 1	0.9192	0.8928	0.9102	0.9214	0.9079	0.9738	0.9738	0.9602
Ejecución 2	0.9192	0.8928	0.9345	0.9214	0.4991	0.9079	0.9738	0.9602
Ejecución 3	0.9192	0.8928	0.9345	0.9214	0.4991	0.9079	0.9738	0.9602
Ejecución 4	0.9192	0.8928	0.9345	0.9214	0.4991	0.9079	0.9738	0.9602
Ejecución 5	0.9192	0.8928	0.9237	0.9214	0.9079	0.9079	0.9345	0.9602
Ejecución 6	0.9192	0.8928	0.9237	0.9214	0.9079	0.9079	0.9345	0.9602
Ejecución 7	0.9192	0.8928	0.9237	0.9214	0.9079	0.9079	0.9345	0.9602
Ejecución 8	0.9192	0.8928	0.9237	0.9214	0.9079	0.9079	0.9345	0.9602
Ejecución 9	0.9192	0.8928	0.9237	0.9214	0.9079	0.9079	0.9345	0.9602
Ejecución 10	0.9192	0.8928	0.9102	0.9214	0.9079	0.9079	0.9738	0.9602
Page_blocks	0.0569 ± 0.0	0.0564 ± 0.0	0.9807 ± 0.0156	0.9855 ± 0.001	0.994 ± 0.0006	0.9939 ± 0.0006	0.9969 ± 0.002	0.9964 ± 0.0022
Ejecución 1	0.0569	0.0564	0.9869	0.9876	0.994	0.9938	0.9971	0.9976
Ejecución 2	0.0569	0.0564	0.9339	0.9852	0.9946	0.994	0.999	0.9977
Ejecución 3	0.0569	0.0564	0.9856	0.9847	0.9938	0.9934	0.9974	0.9974
Ejecución 4	0.0569	0.0564	0.9854	0.9868	0.993	0.9936	0.9975	0.9905
Ejecución 5	0.0569	0.0564	0.9864	0.9854	0.9934	0.9946	0.9913	0.9952
Ejecución 6	0.0569	0.0564	0.9846	0.9852	0.9941	0.9945	0.9961	0.9974
Ejecución 7	0.0569	0.0564	0.9844	0.9852	0.9938	0.994	0.997	0.9974
Ejecución 8	0.0569	0.0564	0.9866	0.9844	0.9938	0.9926	0.9973	0.9952
Ejecución 9	0.0569	0.0564	0.9864	0.9856	0.9952	0.9938	0.9977	0.9976
Ejecución 10	0.0569	0.0564	0.9868	0.9845	0.9947	0.9945	0.9986	0.9979
Phoneme	0.0016 ± 0.0	0.0016 ± 0.0	0.2822 ± 0.3581	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	0.0016	0.0016	0.6603	1.0	1.0	1.0	1.0	1.0
Ejecución 2	0.0016	0.0016	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 3	0.0016	0.0016	0.0052	1.0	1.0	1.0	1.0	1.0
Ejecución 4	0.0016	0.0016	0.0087	1.0	1.0	1.0	1.0	1.0
Ejecución 5	0.0016	0.0016	0.0104	1.0	1.0	1.0	1.0	1.0
Ejecución 6	0.0016	0.0016	0.6603	1.0	1.0	1.0	1.0	1.0
Ejecución 7	0.0016	0.0016	0.4545	1.0	1.0	1.0	1.0	1.0
Ejecución 8	0.0016	0.0016	0.0087	1.0	1.0	1.0	1.0	1.0
Ejecución 9	0.0016	0.0016	0.0052	1.0	1.0	1.0	1.0	1.0
Ejecución 10	0.0016	0.0016	0.0087	1.0	1.0	1.0	1.0	1.0
Pima	0.0992 ± 0.0	0.0992 ± 0.0	0.7388 ± 0.2114	0.9989 ± 0.0021	1.0 ± 0.0	0.8005 ± 0.3991	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	0.0992	0.0992	0.5496	1.0	1.0	1.0	1.0	1.0
Ejecución 2	0.0992	0.0992	0.9947	1.0	1.0	1.0	1.0	1.0
Ejecución 3	0.0992	0.0992	0.9947	1.0	1.0	1.0	1.0	1.0

Continúa en la siguiente página

Tabla A.4 – continúa de la página anterior								
Nombre	0% restricciones		10% restricciones		15% restricciones		20% restricciones	
	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D
Ejecución 4	0.0992	0.0992	0.0992	1.0	0.9947	1.0	1.0	1.0
Ejecución 5	0.0992	0.0992	0.5496	1.0	1.0	1.0	1.0	1.0
Ejecución 6	0.0992	0.0992	0.5496	0.9947	1.0	0.0023	1.0	1.0
Ejecución 7	0.0992	0.0992	0.5496	1.0	1.0	1.0	1.0	1.0
Ejecución 8	0.0992	0.0992	0.5496	0.9947	1.0	0.0023	1.0	1.0
Ejecución 9	0.0992	0.0992	0.6611	1.0	1.0	1.0	1.0	1.0
Ejecución 10	0.0992	0.0992	0.9947	1.0	1.0	1.0	1.0	1.0
Saheart	0.0051 ± 0.0	0.0051 ± 0.0	0.5443 ± 0.2157	0.2819 ± 0.0502	0.6446 ± 0.3745	0.5967 ± 0.3556	0.9366 ± 0.1903	1.0 ± 0.0
Ejecución 1	0.0051	0.0051	0.6542	0.33	0.9311	0.9311	1.0	1.0
Ejecución 2	0.0051	0.0051	0.0139	0.2507	0.9567	0.4025	1.0	1.0
Ejecución 3	0.0051	0.0051	0.6542	0.33	0.9311	0.9311	1.0	1.0
Ejecución 4	0.0051	0.0051	0.3553	0.2087	0.0155	0.0523	0.3657	1.0
Ejecución 5	0.0051	0.0051	0.6542	0.33	0.0155	0.9311	1.0	1.0
Ejecución 6	0.0051	0.0051	0.7559	0.2087	0.9567	0.0523	1.0	1.0
Ejecución 7	0.0051	0.0051	0.4727	0.33	0.3299	0.9311	1.0	1.0
Ejecución 8	0.0051	0.0051	0.4727	0.33	0.8397	0.9311	1.0	1.0
Ejecución 9	0.0051	0.0051	0.6542	0.2507	0.5128	0.4025	1.0	1.0
Ejecución 10	0.0051	0.0051	0.7559	0.2507	0.9567	0.4025	1.0	1.0
Satimage	-0.0008 ± 0.0001	-0.0012 ± 0.0002	0.722 ± 0.2357	0.4495 ± 0.3266	0.9934 ± 0.0074	0.9971 ± 0.0059	0.9988 ± 0.0006	0.9991 ± 0.0003
Ejecución 1	-0.0007	-0.0013	0.8539	0.1429	0.9993	0.9992	0.9996	0.9993
Ejecución 2	-0.0007	-0.0014	0.053	0.0417	0.9812	0.9996	0.9996	0.9992
Ejecución 3	-0.0008	-0.001	0.7065	0.6812	0.9992	0.9996	0.9981	0.9989
Ejecución 4	-0.0009	-0.0011	0.8006	0.8172	0.9945	0.9988	0.9986	0.9996
Ejecución 5	-0.0009	-0.0011	0.8712	0.1619	0.9796	0.9971	0.9974	0.9989
Ejecución 6	-0.0007	-0.0013	0.8704	0.4655	0.9868	0.999	0.9992	0.9992
Ejecución 7	-0.0007	-0.001	0.6824	0.3983	0.9985	0.9996	0.9986	0.9992
Ejecución 8	-0.0007	-0.0012	0.8435	0.0365	0.9985	0.9993	0.9988	0.9986
Ejecución 9	-0.0007	-0.0009	0.8665	0.8774	0.9982	0.9796	0.9989	0.999
Ejecución 10	-0.0007	-0.0011	0.6718	0.872	0.9985	0.999	0.9988	0.9996
Segment	0.0006 ± 0.0	0.0001 ± 0.0	0.0117 ± 0.0027	0.0209 ± 0.0006	0.0523 ± 0.0181	0.0584 ± 0.0	0.9656 ± 0.0437	0.7794 ± 0.0
Ejecución 1	0.0006	0.0001	0.0161	0.0219	0.0431	0.0584	0.9966	0.7794
Ejecución 2	0.0006	0.0001	0.0116	0.0205	0.0536	0.0584	0.983	0.7794
Ejecución 3	0.0006	0.0001	0.012	0.0219	0.0203	0.0584	0.8741	0.7794
Ejecución 4	0.0006	0.0001	0.0099	0.0205	0.0667	0.0584	0.983	0.7794
Ejecución 5	0.0006	0.0001	0.0157	0.0205	0.0885	0.0584	0.9932	0.7794
Ejecución 6	0.0006	0.0001	0.0125	0.0205	0.0454	0.0584	0.8837	0.7794
Ejecución 7	0.0006	0.0001	0.0064	0.0205	0.0378	0.0584	0.9831	0.7794
Ejecución 8	0.0006	0.0001	0.0125	0.0205	0.0407	0.0584	0.983	0.7794
Ejecución 9	0.0006	0.0001	0.0098	0.0219	0.0654	0.0584	0.9832	0.7794
Ejecución 10	0.0006	0.0001	0.0103	0.0205	0.0613	0.0584	0.9932	0.7794
Sonar	0.7146 ± 0.0	0.6983 ± 0.0	0.6698 ± 0.0233	0.7815 ± 0.0	0.0059 ± 0.0041	0.0029 ± 0.0	0.0019 ± 0.0007	0.0014 ± 0.0
Ejecución 1	0.7146	0.6983	0.6507	0.7815	0.0067	0.0029	0.0014	0.0014
Ejecución 2	0.7146	0.6983	0.6507	0.7815	0.0111	0.0029	0.003	0.0014
Ejecución 3	0.7146	0.6983	0.6507	0.7815	0.0111	0.0029	0.003	0.0014
Ejecución 4	0.7146	0.6983	0.6983	0.7815	0.0014	0.0029	0.0014	0.0014
Ejecución 5	0.7146	0.6983	0.6983	0.7815	0.0014	0.0029	0.0014	0.0014
Ejecución 6	0.7146	0.6983	0.6507	0.7815	0.0111	0.0029	0.003	0.0014
Ejecución 7	0.7146	0.6983	0.6507	0.7815	0.0067	0.0029	0.0014	0.0014
Ejecución 8	0.7146	0.6983	0.6507	0.7815	0.0067	0.0029	0.0014	0.0014
Ejecución 9	0.7146	0.6983	0.6983	0.7815	0.0014	0.0029	0.0014	0.0014
Ejecución 10	0.7146	0.6983	0.6983	0.7815	0.0014	0.0029	0.0014	0.0014
Soybean	0.4263 ± 0.0	0.4199 ± 0.0	0.4278 ± 0.0186	0.4398 ± 0.0	0.3683 ± 0.1395	0.5408 ± 0.0234	0.3457 ± 0.0609	0.3653 ± 0.0706
Ejecución 1	0.4263	0.4199	0.4049	0.4398	0.2743	0.5295	0.2821	0.4227
Ejecución 2	0.4263	0.4199	0.443	0.4398	0.3962	0.5666	0.4173	0.4227
Ejecución 3	0.4263	0.4199	0.443	0.4398	0.7565	0.5295	0.3191	0.4227
Ejecución 4	0.4263	0.4199	0.4049	0.4398	0.368	0.5666	0.378	0.4227
Ejecución 5	0.4263	0.4199	0.4049	0.4398	0.2641	0.5295	0.2436	0.2743
Ejecución 6	0.4263	0.4199	0.443	0.4398	0.3429	0.5295	0.4173	0.2743
Ejecución 7	0.4263	0.4199	0.443	0.4398	0.2641	0.5666	0.2847	0.2743
Ejecución 8	0.4263	0.4199	0.443	0.4398	0.3962	0.4939	0.3191	0.2935
Ejecución 9	0.4263	0.4199	0.443	0.4398	0.2641	0.5666	0.4173	0.4227
Ejecución 10	0.4263	0.4199	0.4049	0.4398	0.3562	0.5295	0.378	0.4227
Spambase	-0.0024 ± 0.0	-0.0024 ± 0.0	0.6967 ± 0.4544	0.0003 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	-0.0024	-0.0024	0.9942	0.0003	1.0	1.0	1.0	1.0
Ejecución 2	-0.0024	-0.0024	0.9942	0.0003	1.0	1.0	1.0	1.0
Ejecución 3	-0.0024	-0.0024	0.9942	0.0003	1.0	1.0	1.0	1.0
Ejecución 4	-0.0024	-0.0024	0.9942	0.0003	1.0	1.0	1.0	1.0
Ejecución 5	-0.0024	-0.0024	0.9942	0.0003	1.0	1.0	1.0	1.0
Ejecución 6	-0.0024	-0.0024	0.9942	0.0003	1.0	1.0	1.0	1.0
Ejecución 7	-0.0024	-0.0024	0.9942	0.0003	1.0	1.0	1.0	1.0
Ejecución 8	-0.0024	-0.0024	0.0026	0.0003	1.0	1.0	1.0	1.0
Ejecución 9	-0.0024	-0.0024	0.0026	0.0003	1.0	1.0	1.0	1.0
Ejecución 10	-0.0024	-0.0024	0.0026	0.0003	1.0	1.0	1.0	1.0
Spectfheart	-0.0896 ± 0.0	-0.0896 ± 0.0	0.3328 ± 0.2359	0.5757 ± 0.0	0.8842 ± 0.0312	0.9421 ± 0.0204	0.2078 ± 0.2642	0.2832 ± 0.3584
Ejecución 1	-0.0896	-0.0896	-0.0247	0.5757	0.9319	0.9319	1.0	0.104
Ejecución 2	-0.0896	-0.0896	0.4872	0.5757	0.8638	0.9319	0.1243	0.104
Ejecución 3	-0.0896	-0.0896	-0.0289	0.5757	0.9319	0.9319	0.104	0.104
Ejecución 4	-0.0896	-0.0896	0.4872	0.5757	0.8638	0.9319	0.1243	0.104
Ejecución 5	-0.0896	-0.0896	0.4872	0.5757	0.8638	0.983	0.1243	1.0
Ejecución 6	-0.0896	-0.0896	0.4872	0.5757	0.8638	0.9319	0.1243	0.104
Ejecución 7	-0.0896	-0.0896	0.4872	0.5757	0.8638	0.9319	0.1243	0.104
Ejecución 8	-0.0896	-0.0896	0.4872	0.5757	0.8638	0.983	0.1243	1.0
Ejecución 9	-0.0896	-0.0896	0.4872	0.5757	0.8638	0.9319	0.1243	0.104
Ejecución 10	-0.0896	-0.0896	-0.0289	0.5757	0.9319	0.9319	0.104	0.104
Tae	0.0135 ± 0.0	0.0219 ± 0.0	0.06 ± 0.0163	-0.0086 ± 0.0	0.0665 ± 0.0234	0.1365 ± 0.0	0.4418 ± 0.0652	0.3679 ± 0.0
Ejecución 1	0.0135	0.0219	0.0212	-0.0086	0.1211	0.1365	0.5509	0.3679
Ejecución 2	0.0135	0.0219	0.0392	-0.0086	0.0338	0.1365	0.3832	0.3679
Ejecución 3	0.0135	0.0219	0.0528	-0.0086	0.0417	0.1365	0.5466	0.3679
Ejecución 4	0.0135	0.0219	0.0731	-0.0086	0.0839	0.1365	0.3832	0.3679
Ejecución 5	0.0135	0.0219	0.0682	-0.0086	0.0608	0.1365	0.4064	0.3679
Ejecución 6	0.0135	0.0219	0.0682	-0.0086	0.0608	0.1365	0.4064	0.3679

Continúa en la siguiente página

A. Resultados experimentales completos.

Tabla A.4 – continúa de la página anterior

Nombre	0 % restricciones		10 % restricciones		15 % restricciones		20 % restricciones	
	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D
Ejecución 7	0.0135	0.0219	0.0682	-0.0086	0.0572	0.1365	0.522	0.3679
Ejecución 8	0.0135	0.0219	0.0731	-0.0086	0.0839	0.1365	0.4066	0.3679
Ejecución 9	0.0135	0.0219	0.0682	-0.0086	0.0608	0.1365	0.4064	0.3679
Ejecución 10	0.0135	0.0219	0.0682	-0.0086	0.0608	0.1365	0.4064	0.3679
Thyroid	-0.0167 ± 0.0	-0.0167 ± 0.0	0.8003 ± 0.2651	0.9905 ± 0.0024	0.8692 ± 0.1037	0.8557 ± 0.0903	0.9987 ± 0.0021	0.9989 ± 0.0023
Ejecución 1	-0.0167	-0.0167	0.9831	0.9888	0.9976	0.7784	1.0	1.0
Ejecución 2	-0.0167	-0.0167	0.7438	0.9942	0.7682	0.8083	0.9946	1.0
Ejecución 3	-0.0167	-0.0167	0.9927	0.9894	0.9841	0.7884	1.0	1.0
Ejecución 4	-0.0167	-0.0167	0.7438	0.9888	0.7682	0.9906	0.9959	1.0
Ejecución 5	-0.0167	-0.0167	0.8858	0.9888	0.9976	0.8034	1.0	0.9944
Ejecución 6	-0.0167	-0.0167	0.8559	0.9942	0.8172	0.8083	1.0	1.0
Ejecución 7	-0.0167	-0.0167	0.9907	0.9888	0.8081	0.8034	1.0	0.9944
Ejecución 8	-0.0167	-0.0167	0.9887	0.9942	0.7832	0.9976	0.9963	1.0
Ejecución 9	-0.0167	-0.0167	0.0644	0.9894	0.9994	0.7884	1.0	1.0
Ejecución 10	-0.0167	-0.0167	0.7537	0.9888	0.7682	0.9906	1.0	1.0
Titanic	0.0032 ± 0.0	0.0032 ± 0.0	0.377 ± 0.3352	0.9877 ± 0.0	0.6423 ± 0.4489	0.9988 ± 0.0037	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	0.0032	0.0032	0.4867	0.9877	1.0	0.9877	1.0	1.0
Ejecución 2	0.0032	0.0032	0.4867	0.9877	1.0	1.0	1.0	1.0
Ejecución 3	0.0032	0.0032	0.9877	0.9877	0.9939	1.0	1.0	1.0
Ejecución 4	0.0032	0.0032	0.0961	0.9877	1.0	1.0	1.0	1.0
Ejecución 5	0.0032	0.0032	0.1089	0.9877	0.0099	1.0	1.0	1.0
Ejecución 6	0.0032	0.0032	0.9877	0.9877	0.4057	1.0	1.0	1.0
Ejecución 7	0.0032	0.0032	0.1059	0.9877	1.0	1.0	1.0	1.0
Ejecución 8	0.0032	0.0032	0.1591	0.9877	0.0099	1.0	1.0	1.0
Ejecución 9	0.0032	0.0032	0.1591	0.9877	0.0099	1.0	1.0	1.0
Ejecución 10	0.0032	0.0032	0.1915	0.9877	0.9939	1.0	1.0	1.0
Vehicle	0.001 ± 0.0	0.001 ± 0.0	0.0886 ± 0.0304	0.0543 ± 0.0205	0.9521 ± 0.0609	0.9735 ± 0.035	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	0.001	0.001	0.1271	0.0588	0.8181	0.9906	1.0	1.0
Ejecución 2	0.001	0.001	0.0574	0.0632	0.8463	0.984	1.0	1.0
Ejecución 3	0.001	0.001	0.1564	0.0264	0.9814	0.9685	1.0	1.0
Ejecución 4	0.001	0.001	0.0608	0.0993	0.9936	0.9814	1.0	1.0
Ejecución 5	0.001	0.001	0.1063	0.0596	0.9872	0.9937	1.0	1.0
Ejecución 6	0.001	0.001	0.0705	0.0399	0.9605	0.8705	1.0	1.0
Ejecución 7	0.001	0.001	0.0812	0.0244	0.9785	0.9814	1.0	1.0
Ejecución 8	0.001	0.001	0.0855	0.06	0.9872	0.9907	1.0	1.0
Ejecución 9	0.001	0.001	0.0652	0.0633	0.9905	0.9872	1.0	1.0
Ejecución 10	0.001	0.001	0.0752	0.0476	0.978	0.9873	1.0	1.0
Wdbc	0.009 ± 0.0	0.009 ± 0.0	0.3667 ± 0.0955	0.6779 ± 0.0908	0.988 ± 0.0032	0.9901 ± 0.0047	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	0.009	0.009	0.1415	0.7319	0.9929	0.9929	1.0	1.0
Ejecución 2	0.009	0.009	0.3006	0.6388	0.9859	0.9929	1.0	1.0
Ejecución 3	0.009	0.009	0.4622	0.6388	0.9929	0.9929	1.0	1.0
Ejecución 4	0.009	0.009	0.3645	0.9374	0.9859	0.9789	1.0	1.0
Ejecución 5	0.009	0.009	0.4665	0.6388	0.9859	0.9929	1.0	1.0
Ejecución 6	0.009	0.009	0.3532	0.6388	0.9859	0.9859	1.0	1.0
Ejecución 7	0.009	0.009	0.3006	0.6388	0.9859	0.9929	1.0	1.0
Ejecución 8	0.009	0.009	0.4622	0.6388	0.9929	0.9859	1.0	1.0
Ejecución 9	0.009	0.009	0.408	0.6388	0.9859	0.9929	1.0	1.0
Ejecución 10	0.009	0.009	0.408	0.6388	0.9859	0.9929	1.0	1.0
Wine	0.649 ± 0.0	0.661 ± 0.0	0.6855 ± 0.0225	0.5211 ± 0.0	0.8348 ± 0.0026	0.8858 ± 0.0	0.7557 ± 0.065	0.8033 ± 0.0
Ejecución 1	0.649	0.661	0.713	0.5211	0.8373	0.8858	0.7868	0.8033
Ejecución 2	0.649	0.661	0.6672	0.5211	0.8373	0.8858	0.7868	0.8033
Ejecución 3	0.649	0.661	0.713	0.5211	0.8373	0.8858	0.5621	0.8033
Ejecución 4	0.649	0.661	0.6672	0.5211	0.8373	0.8858	0.7868	0.8033
Ejecución 5	0.649	0.661	0.713	0.5211	0.8322	0.8858	0.7699	0.8033
Ejecución 6	0.649	0.661	0.6672	0.5211	0.8322	0.8858	0.7694	0.8033
Ejecución 7	0.649	0.661	0.6672	0.5211	0.8322	0.8858	0.7694	0.8033
Ejecución 8	0.649	0.661	0.6672	0.5211	0.8322	0.8858	0.7694	0.8033
Ejecución 9	0.649	0.661	0.6672	0.5211	0.8373	0.8858	0.7868	0.8033
Ejecución 10	0.649	0.661	0.713	0.5211	0.8322	0.8858	0.7699	0.8033
Zoo	0.0173 ± 0.0	-0.0355 ± 0.0	0.0093 ± 0.0133	-0.0176 ± 0.0215	0.0591 ± 0.0186	0.0366 ± 0.0221	0.1088 ± 0.0313	0.127 ± 0.0443
Ejecución 1	0.0173	-0.0355	-0.0037	-0.0196	0.0676	0.0633	0.0859	0.1928
Ejecución 2	0.0173	-0.0355	0.0253	-0.034	0.0768	0.0102	0.135	0.0468
Ejecución 3	0.0173	-0.0355	-0.0037	-0.0215	0.0327	0.039	0.0859	0.1384
Ejecución 4	0.0173	-0.0355	0.0216	-0.0261	0.0548	0.0024	0.1532	0.1554
Ejecución 5	0.0173	-0.0355	-0.0037	-0.0215	0.0327	0.039	0.0859	0.1384
Ejecución 6	0.0173	-0.0355	0.0135	-0.034	0.0639	0.039	0.1329	0.0468
Ejecución 7	0.0173	-0.0355	0.0253	-0.0215	0.0768	0.039	0.135	0.1156
Ejecución 8	0.0173	-0.0355	-0.0031	-0.0215	0.0764	0.039	0.0534	0.1384
Ejecución 9	0.0173	-0.0355	-0.0037	0.0451	0.0327	0.0795	0.0859	0.1585
Ejecución 10	0.0173	-0.0355	0.0253	-0.0215	0.0768	0.0157	0.135	0.1384
Media	0.1616 ± 0.2879	0.1543 ± 0.2806	0.4878 ± 0.3101	0.5478 ± 0.3643	0.7162 ± 0.345	0.757 ± 0.3459	0.789 ± 0.3154	0.8228 ± 0.3018

Tabla A.5.: Resultados de las ejecuciones del SCCSOM-EAL-SL.

Nombre	0 % restricciones		10 % restricciones		15 % restricciones		20 % restricciones	
	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D
Appendicitis	0.7262 ± 0.0	0.8737 ± 0.0	0.6132 ± 0.3287	0.894 ± 0.0202	0.3253 ± 0.3769	0.7097 ± 0.2322	0.8924 ± 0.1916	0.9607 ± 0.0131
Ejecución 1	0.7262	0.8737	0.8284	0.8737	1.0	0.761	0.9563	0.9563
Ejecución 2	0.7262	0.8737	0.1111	0.9142	0.4667	0.2701	0.9563	0.9563
Ejecución 3	0.7262	0.8737	0.8284	0.9142	-0.005	0.761	0.9563	0.9563
Ejecución 4	0.7262	0.8737	0.8284	0.8737	0.0045	0.8759	0.3175	0.9563
Ejecución 5	0.7262	0.8737	0.8284	0.8737	0.183	0.761	0.9563	0.9563
Ejecución 6	0.7262	0.8737	0.1111	0.9142	0.4667	0.761	0.9563	0.9563
Ejecución 7	0.7262	0.8737	0.8284	0.9142	0.0712	0.761	0.9563	1.0
Ejecución 8	0.7262	0.8737	0.1111	0.8737	-0.005	1.0	0.9563	0.9563
Ejecución 9	0.7262	0.8737	0.8284	0.9142	0.0712	0.8759	0.9563	0.9563
Ejecución 10	0.7262	0.8737	0.8284	0.8737	1.0	0.2701	0.9563	0.9563
Balance	0.1999 ± 0.0	0.1999 ± 0.0	0.6857 ± 0.0889	0.7619 ± 0.065	0.9965 ± 0.0	0.9968 ± 0.0029	0.9874 ± 0.0334	1.0 ± 0.0
Ejecución 1	0.1999	0.1999	0.6022	0.762	0.9965	0.993	1.0	1.0
Ejecución 2	0.1999	0.1999	0.6022	0.762	0.9965	0.9965	1.0	1.0
Ejecución 3	0.1999	0.1999	0.6649	0.7385	0.9965	1.0	1.0	1.0
Ejecución 4	0.1999	0.1999	0.6649	0.8582	0.9965	1.0	0.888	1.0
Ejecución 5	0.1999	0.1999	0.6022	0.6114	0.9965	0.993	1.0	1.0
Ejecución 6	0.1999	0.1999	0.6022	0.762	0.9965	0.9965	1.0	1.0
Ejecución 7	0.1999	0.1999	0.7165	0.8582	0.9965	1.0	1.0	1.0
Ejecución 8	0.1999	0.1999	0.8512	0.7661	0.9965	0.9965	0.9861	1.0
Ejecución 9	0.1999	0.1999	0.7211	0.7385	0.9965	1.0	1.0	1.0
Ejecución 10	0.1999	0.1999	0.8295	0.762	0.9965	0.993	1.0	1.0
Banana	0.995 ± 0.0	0.995 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	0.995	0.995	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 2	0.995	0.995	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 3	0.995	0.995	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 4	0.995	0.995	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 5	0.995	0.995	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 6	0.995	0.995	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 7	0.995	0.995	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 8	0.995	0.995	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 9	0.995	0.995	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 10	0.995	0.995	1.0	1.0	1.0	1.0	1.0	1.0
Breast_cancer	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 3	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 4	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 5	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 6	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 7	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 8	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 10	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Contraceptive	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 3	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 4	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 5	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 6	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 7	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 8	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 10	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ecoli	0.7971 ± 0.0	0.8254 ± 0.0	0.8269 ± 0.0019	0.8173 ± 0.0	0.8996 ± 0.0086	0.8981 ± 0.0022	0.9666 ± 0.0056	0.9444 ± 0.0
Ejecución 1	0.7971	0.8254	0.824	0.8173	0.9041	0.8974	0.9696	0.9444
Ejecución 2	0.7971	0.8254	0.824	0.8173	0.9041	0.8974	0.9696	0.9444
Ejecución 3	0.7971	0.8254	0.824	0.8173	0.9041	0.8974	0.9696	0.9444
Ejecución 4	0.7971	0.8254	0.8281	0.8173	0.8741	0.8974	0.9499	0.9444
Ejecución 5	0.7971	0.8254	0.8281	0.8173	0.9016	0.8974	0.9679	0.9444
Ejecución 6	0.7971	0.8254	0.8281	0.8173	0.9016	0.8974	0.9679	0.9444
Ejecución 7	0.7971	0.8254	0.8281	0.8173	0.9016	0.8974	0.9679	0.9444
Ejecución 8	0.7971	0.8254	0.8281	0.8173	0.9016	0.8974	0.9679	0.9444
Ejecución 9	0.7971	0.8254	0.8281	0.8173	0.9016	0.8974	0.9679	0.9444
Ejecución 10	0.7971	0.8254	0.8281	0.8173	0.9016	0.9048	0.9679	0.9444
Glass	0.4956 ± 0.0	0.5075 ± 0.0	0.2399 ± 0.0524	0.1876 ± 0.0354	0.3595 ± 0.0971	0.508 ± 0.102	0.656 ± 0.0403	0.6158 ± 0.0
Ejecución 1	0.4956	0.5075	0.2196	0.2585	0.5732	0.6621	0.6125	0.6158
Ejecución 2	0.4956	0.5075	0.2285	0.1699	0.2945	0.4347	0.6018	0.6158
Ejecución 3	0.4956	0.5075	0.2175	0.1699	0.4939	0.4347	0.671	0.6158
Ejecución 4	0.4956	0.5075	0.2316	0.1699	0.2887	0.4854	0.6939	0.6158
Ejecución 5	0.4956	0.5075	0.1824	0.1699	0.392	0.4347	0.6138	0.6158
Ejecución 6	0.4956	0.5075	0.2285	0.1699	0.2945	0.4347	0.6018	0.6158
Ejecución 7	0.4956	0.5075	0.1824	0.1699	0.392	0.6621	0.6838	0.6158
Ejecución 8	0.4956	0.5075	0.3387	0.1699	0.2887	0.4347	0.6939	0.6158
Ejecución 9	0.4956	0.5075	0.3387	0.1699	0.2887	0.4347	0.6937	0.6158
Ejecución 10	0.4956	0.5075	0.2316	0.2585	0.2887	0.6621	0.6939	0.6158
Haberman	0.9726 ± 0.0	0.9726 ± 0.0	0.9919 ± 0.0163	0.9878 ± 0.0187	0.9863 ± 0.0	0.9863 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	0.9726	0.9726	1.0	1.0	0.9863	0.9863	1.0	1.0
Ejecución 2	0.9726	0.9726	1.0	0.9593	0.9863	0.9863	1.0	1.0
Ejecución 3	0.9726	0.9726	1.0	1.0	0.9863	0.9863	1.0	1.0
Ejecución 4	0.9726	0.9726	1.0	1.0	0.9863	0.9863	1.0	1.0
Ejecución 5	0.9726	0.9726	0.9593	1.0	0.9863	0.9863	1.0	1.0
Ejecución 6	0.9726	0.9726	1.0	0.9593	0.9863	0.9863	1.0	1.0
Ejecución 7	0.9726	0.9726	0.9593	1.0	0.9863	0.9863	1.0	1.0
Ejecución 8	0.9726	0.9726	1.0	1.0	0.9863	0.9863	1.0	1.0
Ejecución 9	0.9726	0.9726	1.0	0.9593	0.9863	0.9863	1.0	1.0
Ejecución 10	0.9726	0.9726	1.0	1.0	0.9863	0.9863	1.0	1.0
Hayes_roth	0.0248 ± 0.0	0.0058 ± 0.0	0.0222 ± 0.0251	0.0754 ± 0.0411	0.0448 ± 0.0467	0.0533 ± 0.0273	0.4268 ± 0.0938	0.3741 ± 0.1292

Continúa en la siguiente página

A. Resultados experimentales completos.

Tabla A.5 – continúa de la página anterior								
Nombre	0 % restricciones		10 % restricciones		15 % restricciones		20 % restricciones	
	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D
Ejecución 1	0.0248	0.0058	-0.003	0.0404	0.0106	0.0369	0.5413	0.2503
Ejecución 2	0.0248	0.0058	0.0259	0.1228	0.1134	0.0369	0.3875	0.2503
Ejecución 3	0.0248	0.0058	0.0046	0.0759	-0.0162	0.0579	0.417	0.4518
Ejecución 4	0.0248	0.0058	0.0259	0.1049	0.0756	0.0488	0.3681	0.2299
Ejecución 5	0.0248	0.0058	0.0089	-0.0028	0.0256	0.0488	0.4653	0.5145
Ejecución 6	0.0248	0.0058	0.017	0.0837	0.0834	0.0369	0.4036	0.2503
Ejecución 7	0.0248	0.0058	0.0004	0.1049	0.0119	0.0488	0.23	0.5145
Ejecución 8	0.0248	0.0058	0.0098	0.1049	0.0106	0.0488	0.5413	0.5145
Ejecución 9	0.0248	0.0058	0.0837	0.1049	0.1222	0.1326	0.3729	0.5145
Ejecución 10	0.0248	0.0058	0.0487	0.0141	0.0106	0.0369	0.5413	0.2503
Heart	0.9705 ± 0.0	0.9705 ± 0.0	0.9559 ± 0.0145	0.9618 ± 0.0174	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	0.9705	0.9705	0.9705	0.9705	1.0	1.0	1.0	1.0
Ejecución 2	0.9705	0.9705	0.9414	0.9705	1.0	1.0	1.0	1.0
Ejecución 3	0.9705	0.9705	0.9414	0.9705	1.0	1.0	1.0	1.0
Ejecución 4	0.9705	0.9705	0.9705	0.927	1.0	1.0	1.0	1.0
Ejecución 5	0.9705	0.9705	0.9414	0.9705	1.0	1.0	1.0	1.0
Ejecución 6	0.9705	0.9705	0.9705	0.927	1.0	1.0	1.0	1.0
Ejecución 7	0.9705	0.9705	0.9705	0.9705	1.0	1.0	1.0	1.0
Ejecución 8	0.9705	0.9705	0.9705	0.9705	1.0	1.0	1.0	1.0
Ejecución 9	0.9705	0.9705	0.9414	0.9705	1.0	1.0	1.0	1.0
Ejecución 10	0.9705	0.9705	0.9414	0.9705	1.0	1.0	1.0	1.0
Ionosphere	0.9771 ± 0.0	0.9771 ± 0.0	0.9863 ± 0.0209	0.9863 ± 0.0209	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	0.9771	0.9771	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 2	0.9771	0.9771	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 3	0.9771	0.9771	0.9545	0.9545	1.0	1.0	1.0	1.0
Ejecución 4	0.9771	0.9771	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 5	0.9771	0.9771	0.9545	1.0	1.0	1.0	1.0	1.0
Ejecución 6	0.9771	0.9771	1.0	0.9545	1.0	1.0	1.0	1.0
Ejecución 7	0.9771	0.9771	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 8	0.9771	0.9771	1.0	0.9545	1.0	1.0	1.0	1.0
Ejecución 9	0.9771	0.9771	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 10	0.9771	0.9771	0.9545	1.0	1.0	1.0	1.0	1.0
Iris	0.941 ± 0.0	0.9601 ± 0.0	0.7016 ± 0.0875	0.786 ± 0.0104	0.6036 ± 0.1628	0.7896 ± 0.1358	0.7651 ± 0.1838	0.9167 ± 0.0542
Ejecución 1	0.941	0.9601	0.6577	0.7964	0.3989	0.8349	0.7388	0.9603
Ejecución 2	0.941	0.9601	0.6466	0.7757	0.8622	0.8349	0.4076	0.941
Ejecución 3	0.941	0.9601	0.5272	0.7964	0.5791	0.8349	0.8626	0.9603
Ejecución 4	0.941	0.9601	0.7249	0.7964	0.3989	0.8349	0.8441	0.8512
Ejecución 5	0.941	0.9601	0.8082	0.7757	0.4389	0.8349	0.8626	0.9603
Ejecución 6	0.941	0.9601	0.6831	0.7757	0.6571	0.3823	0.9026	0.9799
Ejecución 7	0.941	0.9601	0.8082	0.7757	0.6737	0.8349	0.8626	0.9603
Ejecución 8	0.941	0.9601	0.6267	0.7757	0.8622	0.8349	0.4076	0.8512
Ejecución 9	0.941	0.9601	0.8082	0.7964	0.6571	0.8349	0.8626	0.8512
Ejecución 10	0.941	0.9601	0.7249	0.7964	0.5075	0.8349	0.9002	0.8512
Monk2	0.019 ± 0.0	0.0189 ± 0.0	0.2571 ± 0.0369	0.2871 ± 0.0994	0.9785 ± 0.004	0.979 ± 0.0043	0.9799 ± 0.0031	0.9813 ± 0.0004
Ejecución 1	0.019	0.0189	0.2395	0.1352	0.9729	0.9724	0.9724	0.9816
Ejecución 2	0.019	0.0189	0.2395	0.3522	0.9729	0.9816	0.9768	0.9811
Ejecución 3	0.019	0.0189	0.2395	0.3522	0.9821	0.9821	0.9778	0.9811
Ejecución 4	0.019	0.0189	0.3255	0.1352	0.9811	0.9724	0.9806	0.9806
Ejecución 5	0.019	0.0189	0.2107	0.1352	0.9811	0.9724	0.9827	0.9811
Ejecución 6	0.019	0.0189	0.2395	0.3522	0.9724	0.9816	0.9816	0.9816
Ejecución 7	0.019	0.0189	0.2395	0.3522	0.9821	0.9821	0.9816	0.9816
Ejecución 8	0.019	0.0189	0.3255	0.3522	0.9816	0.9821	0.9811	0.9821
Ejecución 9	0.019	0.0189	0.2721	0.3522	0.9816	0.9816	0.9816	0.9811
Ejecución 10	0.019	0.0189	0.2395	0.3522	0.9772	0.9816	0.9827	0.9816
Newthyroid	0.9467 ± 0.0	0.9412 ± 0.0	0.89 ± 0.0493	0.9372 ± 0.0	0.9213 ± 0.064	0.9787 ± 0.0	0.8203 ± 0.0918	0.984 ± 0.0
Ejecución 1	0.9467	0.9412	0.7964	0.9372	0.8831	0.9787	0.6534	0.984
Ejecución 2	0.9467	0.9412	0.8919	0.9372	0.8427	0.9787	0.8066	0.984
Ejecución 3	0.9467	0.9412	0.8919	0.9372	0.8427	0.9787	0.8066	0.984
Ejecución 4	0.9467	0.9412	0.8919	0.9372	0.8427	0.9787	0.8066	0.984
Ejecución 5	0.9467	0.9412	0.9271	0.9372	0.9838	0.9787	0.8953	0.984
Ejecución 6	0.9467	0.9412	0.9202	0.9372	0.9838	0.9787	0.8953	0.984
Ejecución 7	0.9467	0.9412	0.9372	0.9372	0.9838	0.9787	0.8953	0.984
Ejecución 8	0.9467	0.9412	0.9202	0.9372	0.9838	0.9787	0.8953	0.984
Ejecución 9	0.9467	0.9412	0.9271	0.9372	0.9838	0.9787	0.8953	0.984
Ejecución 10	0.9467	0.9412	0.7964	0.9372	0.8831	0.9787	0.6534	0.984
Page_blocks	0.079 ± 0.0019	0.0207 ± 0.0	0.9849 ± 0.0004	0.9849 ± 0.0	0.9873 ± 0.0011	0.9868 ± 0.0006	0.9955 ± 0.001	0.9951 ± 0.0
Ejecución 1	0.0776	0.0207	0.9855	0.9849	0.9884	0.9868	0.9946	0.9951
Ejecución 2	0.0794	0.0207	0.9849	0.9849	0.9869	0.9868	0.9958	0.9951
Ejecución 3	0.0793	0.0207	0.9849	0.9849	0.9869	0.9872	0.9958	0.9951
Ejecución 4	0.0839	0.0207	0.9849	0.9849	0.985	0.9872	0.9947	0.9951
Ejecución 5	0.0776	0.0207	0.9849	0.9849	0.9869	0.9872	0.996	0.9951
Ejecución 6	0.0776	0.0207	0.9855	0.9849	0.9885	0.9872	0.9946	0.9951
Ejecución 7	0.0776	0.0207	0.9844	0.9849	0.9885	0.9872	0.9946	0.9951
Ejecución 8	0.0776	0.0207	0.9844	0.9849	0.9884	0.9872	0.9946	0.9951
Ejecución 9	0.0793	0.0207	0.9849	0.9849	0.9863	0.9855	0.9972	0.9951
Ejecución 10	0.0802	0.0207	0.9849	0.9849	0.9867	0.9855	0.9972	0.9951
Phoneme	0.9949 ± 0.0	0.9949 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	0.9949	0.9949	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 2	0.9949	0.9949	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 3	0.9949	0.9949	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 4	0.9949	0.9949	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 5	0.9949	0.9949	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 6	0.9949	0.9949	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 7	0.9949	0.9949	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 8	0.9949	0.9949	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 9	0.9949	0.9949	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 10	0.9949	0.9949	1.0	1.0	1.0	1.0	1.0	1.0
Pima	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 3	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Continúa en la siguiente página

Tabla A.5 – continúa de la página anterior								
Nombre	0 % restricciones		10 % restricciones		15 % restricciones		20 % restricciones	
	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D
Ejecución 4	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 5	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 6	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 7	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 8	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 10	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Saheart	0.9826 ± 0.0	0.9826 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	0.9826	0.9826	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 2	0.9826	0.9826	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 3	0.9826	0.9826	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 4	0.9826	0.9826	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 5	0.9826	0.9826	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 6	0.9826	0.9826	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 7	0.9826	0.9826	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 8	0.9826	0.9826	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 9	0.9826	0.9826	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 10	0.9826	0.9826	1.0	1.0	1.0	1.0	1.0	1.0
Satimage	0.0157 ± 0.0001	0.0108 ± 0.0001	0.8477 ± 0.0349	0.8681 ± 0.0126	0.9994 ± 0.0006	0.9978 ± 0.0041	0.9986 ± 0.0008	0.9993 ± 0.0002
Ejecución 1	0.0157	0.0109	0.8715	0.8612	0.9996	0.999	0.9992	0.9992
Ejecución 2	0.0155	0.0109	0.8843	0.8874	0.9996	0.9856	0.9996	0.9992
Ejecución 3	0.0156	0.0108	0.8585	0.8528	0.9977	0.9996	0.998	0.9992
Ejecución 4	0.0157	0.0108	0.8788	0.8806	0.9996	0.9993	0.9984	0.9993
Ejecución 5	0.0157	0.0109	0.8577	0.872	0.9993	0.9996	0.9986	0.9989
Ejecución 6	0.0157	0.0109	0.8136	0.8524	0.9996	0.9985	0.9977	0.9996
Ejecución 7	0.0155	0.0107	0.8857	0.8707	0.9996	0.9993	0.9981	0.9996
Ejecución 8	0.0159	0.0108	0.821	0.8806	0.9996	0.9983	0.9984	0.9993
Ejecución 9	0.0158	0.0108	0.7733	0.8498	0.9996	0.9996	0.9997	0.9993
Ejecución 10	0.016	0.0108	0.8322	0.8732	0.9996	0.999	0.9996	0.9992
Segment	0.0025 ± 0.0	0.0016 ± 0.0003	0.0227 ± 0.0041	0.0261 ± 0.0097	0.0924 ± 0.0349	0.0959 ± 0.0359	0.9471 ± 0.037	0.9759 ± 0.0148
Ejecución 1	0.0025	0.0016	0.0174	0.0189	0.1508	0.1024	0.9499	0.9832
Ejecución 2	0.0025	0.0013	0.0217	0.0379	0.0446	0.0509	0.9966	0.9864
Ejecución 3	0.0025	0.0017	0.0229	0.0187	0.1205	0.0844	0.9202	0.9661
Ejecución 4	0.0025	0.0016	0.0212	0.0226	0.0588	0.0623	0.9325	0.9865
Ejecución 5	0.0025	0.0021	0.0287	0.0189	0.125	0.1024	0.8711	0.9832
Ejecución 6	0.0025	0.0018	0.0206	0.0138	0.0712	0.1346	0.9113	0.9898
Ejecución 7	0.0025	0.001	0.0249	0.0319	0.0752	0.1061	0.9797	0.9831
Ejecución 8	0.0025	0.0013	0.0156	0.0379	0.1014	0.0509	0.9504	0.9767
Ejecución 9	0.0025	0.0022	0.0273	0.0419	0.052	0.0926	0.9831	0.9661
Ejecución 10	0.0025	0.0017	0.0266	0.0185	0.1248	0.1728	0.9763	0.9382
Sonar	0.9242 ± 0.0	0.9242 ± 0.0	0.105 ± 0.0123	0.1091 ± 0.0	0.0006 ± 0.0024	0.0029 ± 0.0	0.1012 ± 0.2996	0.0014 ± 0.0
Ejecución 1	0.9242	0.9242	0.1091	0.1091	-0.0	0.0029	0.0014	0.0014
Ejecución 2	0.9242	0.9242	0.1091	0.1091	-0.0	0.0029	0.0014	0.0014
Ejecución 3	0.9242	0.9242	0.1091	0.1091	0.0029	0.0029	0.0014	0.0014
Ejecución 4	0.9242	0.9242	0.0682	0.1091	0.0067	0.0029	0.0014	0.0014
Ejecución 5	0.9242	0.9242	0.1091	0.1091	-0.0	0.0029	0.0014	0.0014
Ejecución 6	0.9242	0.9242	0.1091	0.1091	-0.0	0.0029	0.0014	0.0014
Ejecución 7	0.9242	0.9242	0.1091	0.1091	-0.0	0.0029	0.0014	0.0014
Ejecución 8	0.9242	0.9242	0.1091	0.1091	-0.0	0.0029	0.0014	0.0014
Ejecución 9	0.9242	0.9242	0.1091	0.1091	-0.0	0.0029	0.0014	0.0014
Ejecución 10	0.9242	0.9242	0.1091	0.1091	-0.0031	0.0029	1.0	0.0014
Soybean	0.466 ± 0.0	0.4744 ± 0.0	0.4989 ± 0.0	0.4141 ± 0.0	0.4468 ± 0.0509	0.332 ± 0.0947	0.4811 ± 0.0554	0.5018 ± 0.0407
Ejecución 1	0.466	0.4744	0.4989	0.4141	0.3929	0.5455	0.514	0.3863
Ejecución 2	0.466	0.4744	0.4989	0.4141	0.5202	0.2753	0.4675	0.514
Ejecución 3	0.466	0.4744	0.4989	0.4141	0.5202	0.2753	0.3269	0.514
Ejecución 4	0.466	0.4744	0.4989	0.4141	0.4438	0.2753	0.4675	0.514
Ejecución 5	0.466	0.4744	0.4989	0.4141	0.4284	0.2753	0.5133	0.514
Ejecución 6	0.466	0.4744	0.4989	0.4141	0.3929	0.2753	0.514	0.514
Ejecución 7	0.466	0.4744	0.4989	0.4141	0.4284	0.3757	0.5133	0.4876
Ejecución 8	0.466	0.4744	0.4989	0.4141	0.5202	0.2753	0.4675	0.514
Ejecución 9	0.466	0.4744	0.4989	0.4141	0.4284	0.472	0.5133	0.5463
Ejecución 10	0.466	0.4744	0.4989	0.4141	0.3929	0.2753	0.514	0.514
Spambase	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 3	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 4	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 5	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 6	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 7	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 8	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 10	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Spectfheart	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 3	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 4	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 5	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 6	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 7	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 8	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 10	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Tae	0.1404 ± 0.0	0.1306 ± 0.0	0.1906 ± 0.0236	0.2775 ± 0.0117	0.2985 ± 0.1168	0.3605 ± 0.1775	0.6395 ± 0.1222	0.6221 ± 0.176
Ejecución 1	0.1404	0.1306	0.1263	0.2894	0.3814	0.5429	0.7002	0.4283
Ejecución 2	0.1404	0.1306	0.2	0.2831	0.3859	0.4066	0.5786	0.5505
Ejecución 3	0.1404	0.1306	0.2	0.2894	0.3859	0.5429	0.5786	0.4283
Ejecución 4	0.1404	0.1306	0.1787	0.2747	0.3859	0.0985	0.5786	0.7349
Ejecución 5	0.1404	0.1306	0.1875	0.2894	0.1671	0.5429	0.8644	0.4283
Ejecución 6	0.1404	0.1306	0.2053	0.2747	0.178	0.0985	0.6132	0.7349

Continúa en la siguiente página

A. Resultados experimentales completos.

Tabla A.5 – continúa de la página anterior								
Nombre	0 % restricciones		10 % restricciones		15 % restricciones		20 % restricciones	
	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D	SOM 2D	SOM 3D
Ejecución 7	0.1404	0.1306	0.1875	0.2615	0.1671	0.2168	0.8644	0.8293
Ejecución 8	0.1404	0.1306	0.2	0.2894	0.3859	0.5429	0.5786	0.4283
Ejecución 9	0.1404	0.1306	0.2105	0.2615	0.4295	0.3967	0.5193	0.8293
Ejecución 10	0.1404	0.1306	0.2105	0.2615	0.1176	0.2168	0.5193	0.8293
Thyroid	0.0015 ± 0.0	0.0004 ± 0.0025	0.9892 ± 0.0004	0.9898 ± 0.0014	0.9924 ± 0.0014	0.9931 ± 0.0016	0.999 ± 0.0006	0.9994 ± 0.0002
Ejecución 1	0.0015	-0.0011	0.9896	0.992	0.9925	0.9923	0.9994	0.9994
Ejecución 2	0.0015	0.0051	0.9894	0.9886	0.9925	0.9961	0.9983	0.9994
Ejecución 3	0.0015	-0.0011	0.9891	0.9887	0.9952	0.9923	0.9997	0.9994
Ejecución 4	0.0015	0.0053	0.9889	0.9913	0.9925	0.9923	0.9983	0.9994
Ejecución 5	0.0015	0.0012	0.9888	0.9913	0.9892	0.9923	0.9983	0.9994
Ejecución 6	0.0015	-0.0011	0.9897	0.9887	0.9922	0.9923	0.9994	0.9994
Ejecución 7	0.0015	-0.0011	0.9888	0.9887	0.9922	0.9923	0.9994	0.9994
Ejecución 8	0.0015	-0.0011	0.9888	0.9913	0.9924	0.9923	0.9994	0.9994
Ejecución 9	0.0015	-0.0011	0.9894	0.9887	0.9925	0.9923	0.9983	0.9994
Ejecución 10	0.0015	-0.0011	0.9896	0.9887	0.9925	0.9966	0.9994	0.9989
Titanic	0.9877 ± 0.0	0.9877 ± 0.0	0.9939 ± 0.0	0.9939 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	0.9877	0.9877	0.9939	0.9939	1.0	1.0	1.0	1.0
Ejecución 2	0.9877	0.9877	0.9939	0.9939	1.0	1.0	1.0	1.0
Ejecución 3	0.9877	0.9877	0.9939	0.9939	1.0	1.0	1.0	1.0
Ejecución 4	0.9877	0.9877	0.9939	0.9939	1.0	1.0	1.0	1.0
Ejecución 5	0.9877	0.9877	0.9939	0.9939	1.0	1.0	1.0	1.0
Ejecución 6	0.9877	0.9877	0.9939	0.9939	1.0	1.0	1.0	1.0
Ejecución 7	0.9877	0.9877	0.9939	0.9939	1.0	1.0	1.0	1.0
Ejecución 8	0.9877	0.9877	0.9939	0.9939	1.0	1.0	1.0	1.0
Ejecución 9	0.9877	0.9877	0.9939	0.9939	1.0	1.0	1.0	1.0
Ejecución 10	0.9877	0.9877	0.9939	0.9939	1.0	1.0	1.0	1.0
Vehicle	0.9751 ± 0.0	0.9751 ± 0.0	0.9068 ± 0.0874	0.982 ± 0.005	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	0.9751	0.9751	0.9842	0.9841	1.0	1.0	1.0	1.0
Ejecución 2	0.9751	0.9751	0.9506	0.9841	1.0	1.0	1.0	1.0
Ejecución 3	0.9751	0.9751	0.9659	0.9749	1.0	1.0	1.0	1.0
Ejecución 4	0.9751	0.9751	0.7879	0.9841	1.0	1.0	1.0	1.0
Ejecución 5	0.9751	0.9751	0.7879	0.9841	1.0	1.0	1.0	1.0
Ejecución 6	0.9751	0.9751	0.9497	0.9749	1.0	1.0	1.0	1.0
Ejecución 7	0.9751	0.9751	0.9619	0.9905	1.0	1.0	1.0	1.0
Ejecución 8	0.9751	0.9751	0.9687	0.9841	1.0	1.0	1.0	1.0
Ejecución 9	0.9751	0.9751	0.9619	0.9749	1.0	1.0	1.0	1.0
Ejecución 10	0.9751	0.9751	0.7488	0.9841	1.0	1.0	1.0	1.0
Wdbc	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
Ejecución 1	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 3	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 4	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 5	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 6	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 7	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 8	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Ejecución 10	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Wine	0.7834 ± 0.0	0.771 ± 0.0	0.6893 ± 0.1011	0.7385 ± 0.0	0.9117 ± 0.0711	0.9472 ± 0.0	0.708 ± 0.1498	0.9832 ± 0.0
Ejecución 1	0.7834	0.771	0.7128	0.7385	0.9472	0.9472	0.74	0.9832
Ejecución 2	0.7834	0.771	0.7391	0.7385	0.7696	0.9472	0.9637	0.9832
Ejecución 3	0.7834	0.771	0.7128	0.7385	0.9472	0.9472	0.5361	0.9832
Ejecución 4	0.7834	0.771	0.4911	0.7385	0.9472	0.9472	0.5361	0.9832
Ejecución 5	0.7834	0.771	0.7647	0.7385	0.9472	0.9472	0.8078	0.9832
Ejecución 6	0.7834	0.771	0.7647	0.7385	0.9472	0.9472	0.8078	0.9832
Ejecución 7	0.7834	0.771	0.7391	0.7385	0.7696	0.9472	0.5361	0.9832
Ejecución 8	0.7834	0.771	0.7128	0.7385	0.9472	0.9472	0.8078	0.9832
Ejecución 9	0.7834	0.771	0.7647	0.7385	0.9472	0.9472	0.8078	0.9832
Ejecución 10	0.7834	0.771	0.4911	0.7385	0.9472	0.9472	0.5361	0.9832
Zoo	0.0 ± 0.0	-0.009 ± 0.0	0.0029 ± 0.0118	-0.0009 ± 0.0279	0.0999 ± 0.0517	0.0957 ± 0.063	0.226 ± 0.083	0.269 ± 0.1561
Ejecución 1	0.0	-0.009	0.0152	-0.0009	0.0891	0.0301	0.2116	0.0621
Ejecución 2	0.0	-0.009	0.0139	-0.0192	0.0809	0.2364	0.3322	0.3504
Ejecución 3	0.0	-0.009	-0.0187	-0.007	0.2206	0.0713	0.2637	0.273
Ejecución 4	0.0	-0.009	0.0084	-0.0125	0.0777	0.1057	0.3018	0.449
Ejecución 5	0.0	-0.009	-0.0135	-0.0109	0.1363	0.1057	0.1306	0.0369
Ejecución 6	0.0	-0.009	0.0171	-0.0208	0.1526	0.1208	0.0977	0.449
Ejecución 7	0.0	-0.009	0.0118	0.0809	0.0592	0.1662	0.3018	0.449
Ejecución 8	0.0	-0.009	-0.0033	-0.0069	0.0777	0.0392	0.3018	0.0621
Ejecución 9	0.0	-0.009	-0.0027	-0.0047	0.0669	0.0392	0.2066	0.2685
Ejecución 10	0.0	-0.009	0.001	-0.0069	0.0379	0.0428	0.1126	0.2896
Media	0.6587 ± 0.4099	0.6617 ± 0.4167	0.7227 ± 0.3533	0.744 ± 0.352	0.7724 ± 0.3494	0.7972 ± 0.3355	0.8578 ± 0.2419	0.875 ± 0.2534

Bibliografía

Las referencias se listan por orden alfabético. Aquellas referencias con más de un autor están ordenadas de acuerdo con el primer autor.

- [AJ09] Daniel Duarte Abdala and Xiaoyi Jiang. An evidence accumulation approach to constrained clustering combination. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 361–371. Springer, 2009. [Citado en págs. 59 and 63]
- [Alb09] Rafael Payá Albert. *Gradiente, divergencia y rotacional*. 2009. [No citado]
- [AYH15] Amin Allahyar, Hadi Sadoghi Yazdi, and Ahad Harati. Constrained semi-supervised growing self-organizing map. *Neurocomputing (Amsterdam)*, 147:456–471, Jan 05, 2015. [Citado en pág. 49]
- [CF87] Marie Cottrell and Jean-Claude Fort. Etude d’un processus d’auto-organisation. In *Annales de l’IHP Probabilités et statistiques*, volume 23, pages 1–20, 1987. [Citado en pág. 35]
- [DR05] Ian Davidson and S. S. Ravi. Clustering with constraints: Feasibility issues and the k-means algorithm. In *Proceedings of the 2005 SIAM international conference on data mining*, pages 138–149. SIAM, 2005. [Citado en pág. 38]
- [FJ05] Ana LN Fred and Anil K. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):835–850, 2005. [Citado en pág. 59]
- [GA06] LUIS MIGUEL MERINO GONZALEZ and EVANGELINA SANTOS ALAEZ. *Álgebra Lineal con métodos elementales*. Editorial Paraninfo, 2006. [No citado]
- [GALCG20] Germán González-Almagro, Julián Luengo, José-Ramón Cano, and Salvador García. Dils: constrained clustering through dual iterative local search. *Computers and Operations Research*, 121:104979, 2020. [No citado]
- [Ger79] A. Gersho. Asymptotically optimal block quantization. *IEEE transactions on information theory*, 25(4):373–380, Jul 1979. [Citado en pág. 20]
- [GSW] R. Goebel, J. Siekmann, and W. Wahlster. Lecture notes in artificial intelligence 5632 subseries of lecture notes in computer science. [No citado]
- [Kia01] Melody Y. Kiang. Extending the kohonen self-organizing map networks for clustering analysis. *Computational statistics and data analysis*, 38(2):161–180, 2001. [Citado en pág. 41]
- [Koh82a] Teuvo Kohonen. Analysis of a simple self-organizing process. *Biological cybernetics*, 44(2):135–140, 1982. [No citado]
- [Koh82b] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43(1):59–69, 1982. [Citado en págs. 22 and 35]
- [Koh90] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990. [Citado en pág. 22]
- [Koh01] Teuvo Kohonen. Self-organizing maps, ser. *Information Sciences*. Berlin: Springer, 30, 2001. [No citado]
- [LGA19] Javier León, Palomares Germán, and González Almagro. Self-organizing maps, -05-22 2019. [No citado]
- [Mur95] Fionn Murtagh. Interpreting the kohonen self-organizing feature map using contiguity-constrained clustering. *Pattern Recognition Letters*, 16(4):399–408, 1995. [Citado en pág. 41]

- [Per09] Petra Perner. *Machine Learning and Data Mining in Pattern Recognition: 6th International Conference, MLDM 2009, Leipzig, Germany, July 23-25, 2009, Proceedings*, volume 5632. Springer Science and Business Media, 2009. [No citado]
- [PVG⁺11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. Cited By :23450. [No citado]
- [Ran71] William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971. [No citado]
- [Rom14a] Patricia Román Román. *Estadística descriptiva e introducción a la probabilidad*. Universidad de Granada, 2014. [No citado]
- [Rom14b] Patricia Román Román. *Probabilidad condicionada: teoremas básicos. Independencia de sucesos*. Universidad de Granada, 2014. [No citado]
- [Rom14c] Patricia Román Román. *Variables aleatorias: distribuciones de probabilidad y características*. Universidad de Granada, 2014. [No citado]
- [TGM⁺17] Isaac Triguero, Sergio González, Jose M. Moyano, Salvador García, Jesús Alcalá-Fdez, Julián Luengo, Alberto Fernández, Maria José del Jesús, Luciano Sánchez, and Francisco Herrera. Keel 3.0: an open source software for multi-stage analysis in data mining. *International Journal of Computational Intelligence Systems*, 10(1):1238–1249, 2017. [Citado en pág. 75]
- [WCRS01] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl. Constrained k-means clustering with background knowledge. *ICML*, 1:577–584, 2001. Cited By :52. [Citado en pág. 77]
- [YTLR12] Yan Yang, Wei Tan, Tianrui Li, and Da Ruan. Consensus clustering based on constrained self-organizing map and improved cop-kmeans ensemble in intelligent decision support systems. *Knowledge-based systems*, 32:101–115, Aug 2012. [Citado en págs. 53 and 61]