

UNIVERSIDADE DA CORUÑA

APRENDIZAJE AUTOMÁTICO

TEMA 3:

APRENDIZAJE NO SUPERVISADO



TEMA 3

3.1 Aprendizaje no supervisado: agrupación

3.2 Redes de neuronas no supervisadas



APRENDIZAJE AUTOMÁTICO

INTERÉS

- No hay inteligencia sin aprendizaje (adaptación, mejora, descubrimiento, etc.)
- En la práctica:
 - Exceso de información
 - Escasez de conocimiento
 - Necesidad de automatizar la obtención de conocimiento a partir de información.

APRENDIZAJE AUTOMÁTICO

APLICACIONES

- Minería de datos: uso de datos históricos para mejorar la toma de decisiones
 - Registros médicos -> Conocimiento médico
 - Imágenes del firmamento -> catálogo de objetos estelares
- Aplicaciones software que no se pueden programar con técnicas convencionales
 - Reconocimiento del habla
 - Vehículos autónomos
- Software personalizado
 - Filtro de noticias de interés
 - Gestión de Agenda



APRENDIZAJE AUTOMÁTICO

PROCESO

- Selección del conjunto de datos.
- Análisis de las propiedades de los datos.
- Transformación o preprocesado del conjunto de datos de entrada.
- Seleccionar y aplicar la técnica de aprendizaje automático.
- Extracción de conocimiento.
- Interpretación y evaluación de datos.

APRENDIZAJE AUTOMÁTICO

PROCESO

- ¿Cuándo es mejor usar un modelo de *machine learning* u otro?
- Y después, dentro de ese modelo o técnica, ¿cuál algoritmo es mejor?
- Depende de los datos que tengamos: si tienen ruido, si están etiquetados, si son muchos o pocos, el tiempo que tarda cada modelo, etc.

3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

- **Aprendizaje no supervisado**
 - Sin ningún tipo de señal que indique un comportamiento deseado para el sistema.
 - Son conjuntos de patrones no etiquetados.
- Se desconocen las clases, solo hay grupos similares. El etiquetado de cada muestra individual es impracticable (costoso/imposible recoger información real).
 - Ej. Imágenes de personas ¿cuáles se parecen y cuáles no?
- La **no supervisión** consiste en que el sistema descubra por sí solo características, regularidades, correlaciones o categorías en los datos de entrada y se obtengan a la salida.
- El aprendizaje no supervisado sólo consigue resultados útiles si en los datos de entrada existe cierto grado de redundancia.
 - Sin redundancia sería imposible encontrar patrones o características en los datos.
- No existe un supervisor externo para realizar el aprendizaje.
- Se modifican los parámetros internamente, adaptándose al entorno de la mejor manera posible.



3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

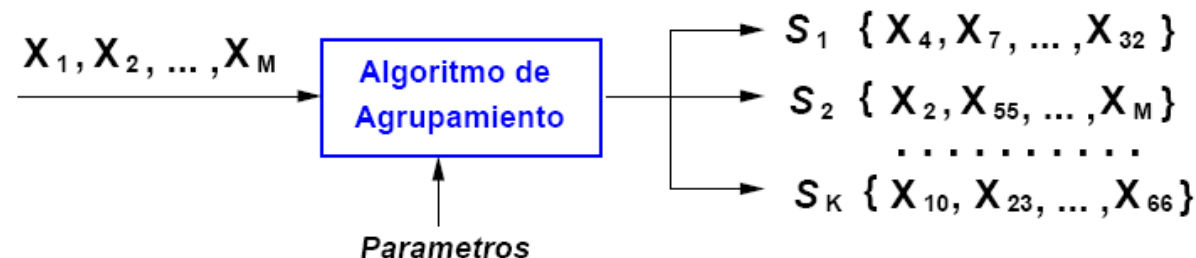
Técnicas de aprendizaje no supervisado

- **Clustering o Agrupación**: se agrupan datos en regiones donde la similitud mutua es elevada, siguiendo algún criterio predefinido.
- **Visualización**: dados unos datos con representación compleja, permitir su visualización.
- **Generación de jerarquías**: dados unos datos en un mismo nivel, generar jerarquías que organicen dichos datos
- **Reducción de la dimensionalidad**: dados unos datos, reducir la dimensión o número de atributos que caracterizan dichos datos
- **Extracción de características**: construyen nuevos atributos (pocos) a partir de los atributos originales (muchos)

3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Clustering o Agrupación

- Se agrupan una serie de datos según criterios habitualmente de distancia; se tratará de disponer los vectores/patrones de entrada de forma que, estén más cercanos aquellos que tengan características comunes.
 - Conjunto de objetos que son "similares" entre ellos y "diferentes" de los objetos que pertenecen a los otros grupos.
 - Ej. Extraer temas de documentos, taxonomía animales/plantas, perfiles de compra, etc.
- A partir de un conjunto de entrada se desea determinar si se puede dividir ese conjunto en diferentes grupos, categorías o *clusters* con un centro o prototipo del cluster.
 - Permite separar datos en grupos (a priori no se sabe cuántos grupos existen).
 - Facilita indicar a qué grupo pertenece cada dato de entrada.
 - Permite caracterizar cada grupo.



3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Clustering o Agrupación

- Cuestiones principales:
 - **Clasificación:** Supervisado. Aprender un método para predecir la clase de una instancia preclasificada.
 - **Clustering** NO supervisado: Encontrar el agrupamiento “natural” de instancias no preclasificadas.
 - ¿Cuántos grupos hay?
 - Pueden ser exclusivos, con solapes, probabilísticos, jerárquicos.
 - ¿En qué orden se procesan los patrones/instancias?
 - ¿Cómo se decide a qué grupo pertenece una nueva instancia?
 - Para decidir si x pertenece a un agrupamiento se necesita: **medida de proximidad o similitud** (medidas de distancia).
 - **Criterio de agrupamiento:** Permite definir cuantitativamente cuándo una partición es mejor que otra.
 - **Algoritmo de agrupamiento:** método que use una medida de similitud con un criterio.

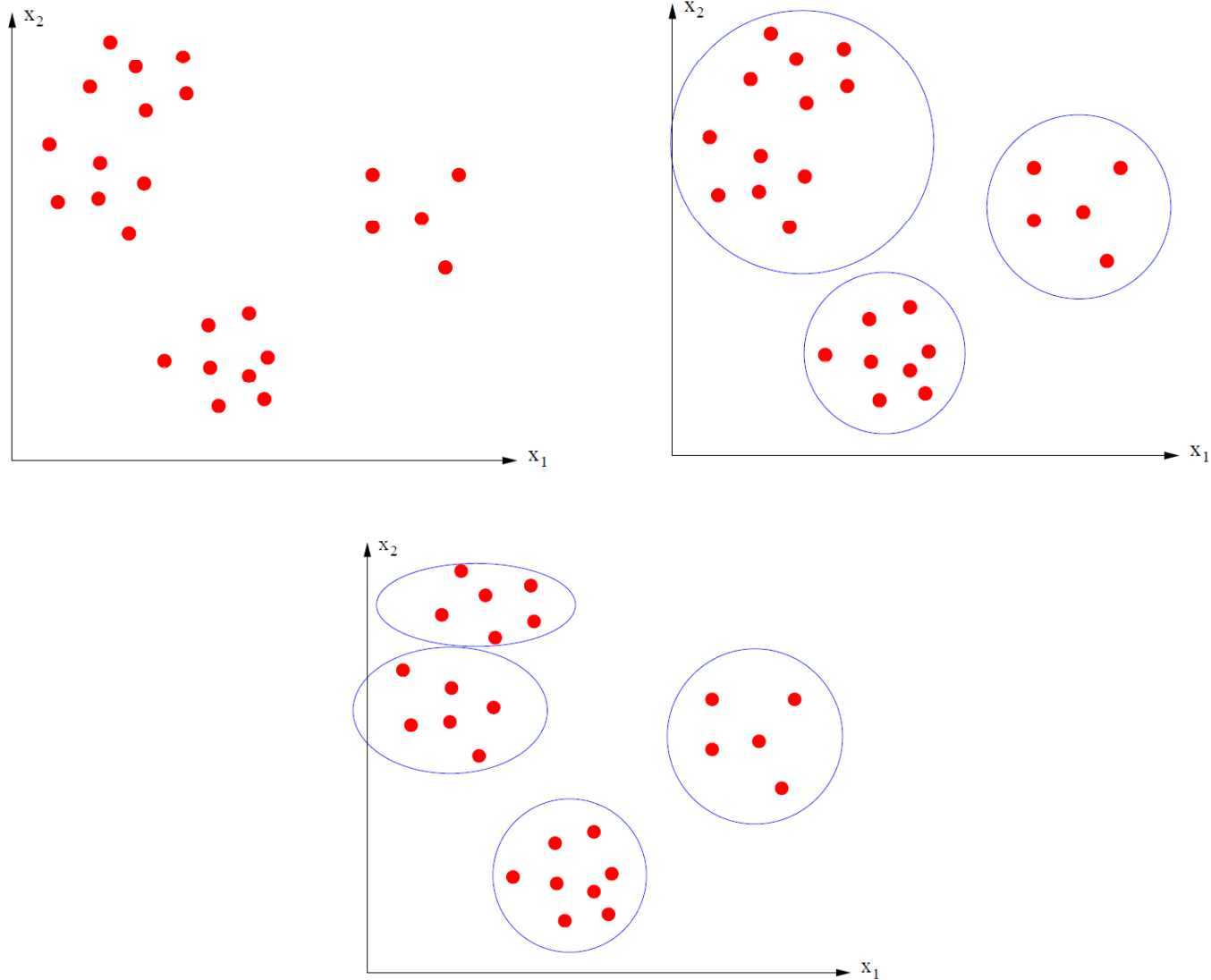
3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Clustering o Agrupación

- **Existen gran cantidad de algoritmos de clustering.**
- Retos:
 - Escalabilidad: normalmente trabajan con pocos datos.
 - Capacidad de manejar diferentes tipos de atributos: numéricos (lo más común), binarios, nominales, ordinales, etc. Se normalizan los diferentes atributos.
 - Clusters de formas arbitrarias: los basados en distancias numéricas tienden a encontrar cluster esféricos.
 - Requerimientos mínimos para especificar parámetros, como el n° de clusters.
 - Manejo de ruido: muchos son sensibles a datos erróneos.
 - Independientes del orden de los datos.
 - Poder funcionar eficientemente con alta dimensionalidad.
 - Que los clusters sean interpretables y utilizables.

3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Clustering o Agrupación



3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Clustering o Agrupación: Tipos de métodos

- **Métodos basados en particiones:** construyen k particiones de los datos, donde cada partición representa un grupo o cluster. Cada grupo tiene al menos un elemento y cada elemento pertenece a un solo grupo.
 - Estos métodos, crean una partición inicial e iteran hasta un criterio de parada. Los más populares son **k-medias** y k-medianas (otros: CLARA y CLARANS).
- **Métodos jerárquicos:** crean descomposiciones de datos siguiendo una jerarquía (e.g., **COBWEB**).
- **Métodos basados en densidades:** Se agrupan objetos mientras su densidad (número de objetos) en la “vecindad” este dentro de un cierto umbral (e.g., DBSCAN, DENCLUE).
- **Métodos basados en rejillas:** se divide el espacio en rejillas a diferentes niveles (e.g, STING, CLIQUE).
- **Métodos basados en modelos:** se encuentra un modelo para cada cluster que mejor ajuste los datos de ese grupo (e.g., COBWEB, AutoClass).

3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Clustering o Agrupación: Tipos de métodos

- Métodos basados en teoría de grafos: utilizan representaciones basadas en grafos (e.g., Chameleon, Delaunay triangulation graph (DTG), highly connected subgraphs (HCS), clustering identification via connectivity kernels (CLICK), cluster affinity search technique (CAST))
- Técnicas basadas en Búsqueda Combinatoria (e.g., Genetically guided algorithm (GGA), TS clustering, SA clustering)
- Técnicas Fuzzy (e.g., Fuzzy c-means (FCM), mountain method (MM), possibilistic c-means clustering algorithm (PCM), fuzzy c-shells (FCS))
- Técnicas basadas en **Redes de Neuronas Artificiales** (e.g., Learning vector quantization (LVQ), self-organizing feature map (**SOM**), **ART**, hyperellipsoidal clustering network (HEC), self-splitting competitive learning network (SPLL))
- Técnicas basadas en Kernels (e.g. Kernel K-means, support vector clustering (SVC))
- Técnicas para Datos Secuenciales (e.g. Similitud secuencial, clustering secuencial indirecto, clustering secuencial estadístico)
- Técnicas para grandes conjuntos de datos (e.g., CLARA, CURE, CLARANS, BIRCH, DBSCAN, DENCLUE, WaveCluster, FC, ART)

3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Clustering o Agrupación no supervisada

- Los algoritmos de clustering **no supervisados** pueden ser según la información a priori:
 - **Paramétricos**: asumen que las densidades de los grupos tienen cierta forma paramétrica conocida, por ejemplo, que los datos siguen una determinada densidad de probabilidad (p.e. Gaussiana), y se reduce a estimar los parámetros.
 - Ej: Algoritmo EM (Expectation-Maximization): particionado, probabilístico y paramétrico
 - **No paramétricos**: no asumen nada sobre el modo en el que se agrupan los objetos. Se basan en agrupar las instancias de entrenamiento siguiendo distintas medidas de distancia/error.
 - Ej: Estrategia Aglomerativa, k-means

3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Clustering o Agrupación no supervisada

- Los algoritmos de clustering **no supervisados** pueden ser según su salida:
 - **Jerárquicos**: jerarquía de grupos en los que unos incluyen a otros. Un grupo raíz contiene a todos los demás. Los más internos serán grupos hoja. Los algoritmos jerárquicos son aquellos en los que se va dividiendo el conjunto de datos por niveles, de modo que en cada nivel generalmente, se unen o se dividen dos grupos del nivel anterior, según si es un algoritmo aglomerativo o divisivo. Los datos se agrupan de manera arborescente:
 - Ej: Estrategia Aglomerativa (dendrograma) Ej. Tipos de plantas, un árbol de especies, etc. COBWEB (probabilístico)
 - **No jerárquicos o de Partición**: generar particiones a un solo nivel. Asumen un conocimiento a priori del n° de clusters en que debe ser dividido el conjunto de datos. Los algoritmos particionales son los que realizan una división inicial de los datos en grupos y luego mueven los objetos de un grupo a otro según se optimice alguna función objetivo/criterio predefinido.
 - Ej: k-means, EM (Expectation-Maximization)

3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Clustering o Agrupación no supervisada

- Diferencias entre métodos jerárquicos y no jerárquicos:
 - Número de clusters:
 - En los métodos jerárquicos podemos obtener una partición con cualquier n° de clusters (partiendo el dendrograma por el punto indicado)
 - En los no jerárquicos el n° de clusters se suele especificar a priori.
 - Pertenencia a un cluster:
 - En los métodos jerárquicos, si se asigna un elemento a un clúster, no puede ser ya reasignado.
 - En los métodos no jerárquicos, los elementos pueden cambiar de cluster en distintas iteraciones.

3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN k-medias (k-means clustering): no jerárquico y no paramétrico

- Técnica iterativa que realiza una discretización o particionamiento del conjunto de datos en **k** grupos indicados por el usuario.
 - La partición debe maximizar la similitud entre los elementos de cada grupo y minimizar la similitud entre diferentes grupos.
- El conjunto de datos contiene **k** agrupamientos **S_i**, cada uno de los cuales puede representarse con su valor medio **μ_i**.
 - **Medida de similitud:** distancia **Euclídea** a μ_i, siendo **n** el n° total de componentes, n° de variables, rasgos o parámetros que tiene cada elemento a agrupar. Ej Ciudad (latitud, longitud).

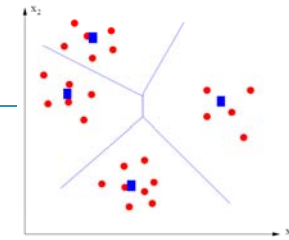
$$d(x_j, \mu_i) = \sqrt{\sum_{k=1}^n (x_{kj} - \mu_{ki})^2}$$

- **Criterio de agrupamiento:** suma total de la distancia cuadrática de cada punto al vector medio de su agrupamiento (mínima varianza).

$$\sum_{i=1}^k \sum_{\mathbf{x}_j \in S_i} \|\mathbf{x}_j - \mu_i\|^2$$

3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

k-medias: no jerárquico y no paramétrico



- **Objetivo:**

- Generar el conjunto de prototipos que minimice una determinada medida de error.
- Encontrar entre todas las particiones que se realicen de los datos en k conjuntos $\{S_i; i = 1, 2, \dots, k\}$ aquella que **minimiza el criterio de agrupamiento**.

- **Ejemplos de Aplicación:**

- Se usa en muchas áreas como segmentación de mercados, visión por computadoras, geoestadística, astronomía, etc.
 - Agrupar usuarios de telefonía móvil.
 - Agrupar pacientes con enfermedades (microarrays de genes)
- También se usa como preprocesamiento para otros algoritmos, por ejemplo para buscar una configuración inicial de grupos.
- Agrupar medicinas: <http://people.revoledu.com/kardi/tutorial/kMean/EjemploNumerico.htm>

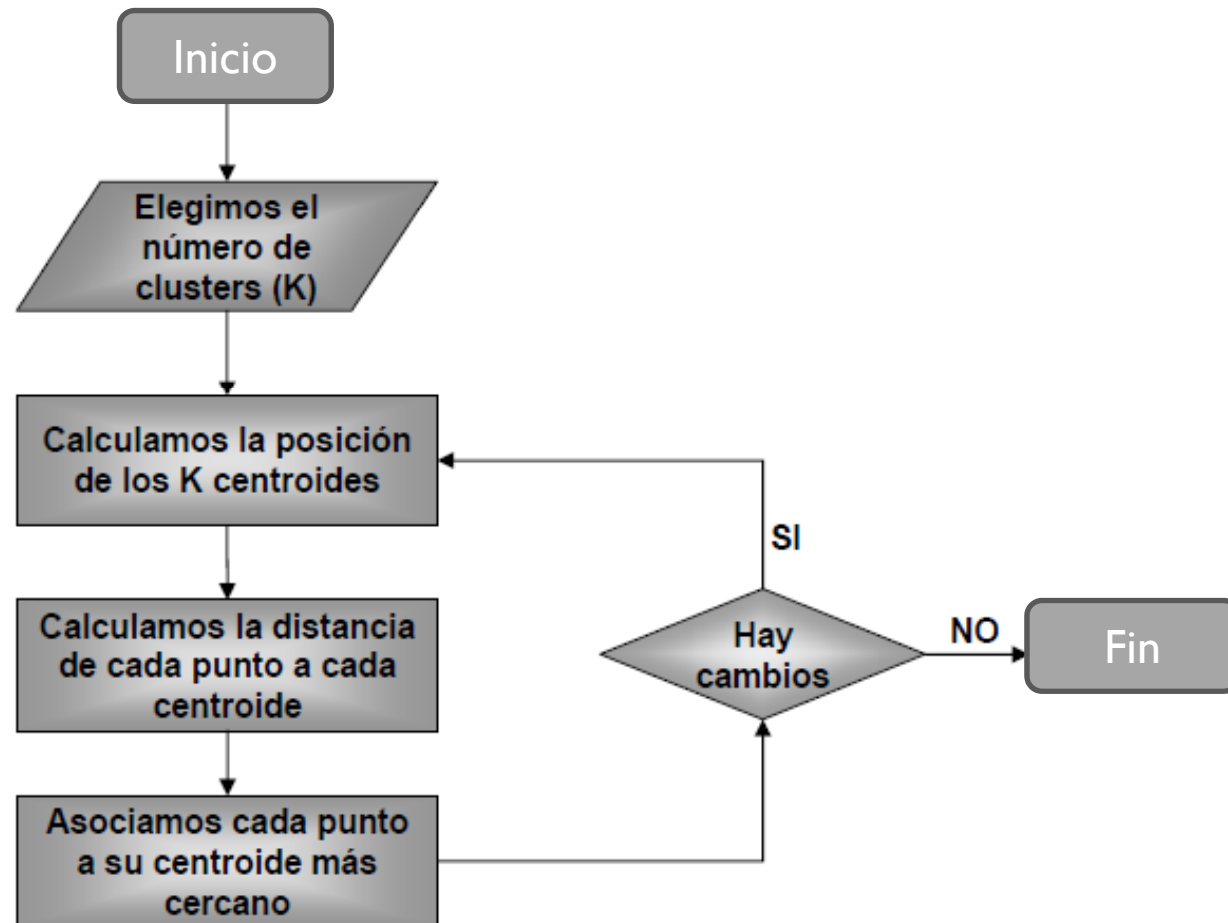
3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

k-medias: no jerárquico y no paramétrico

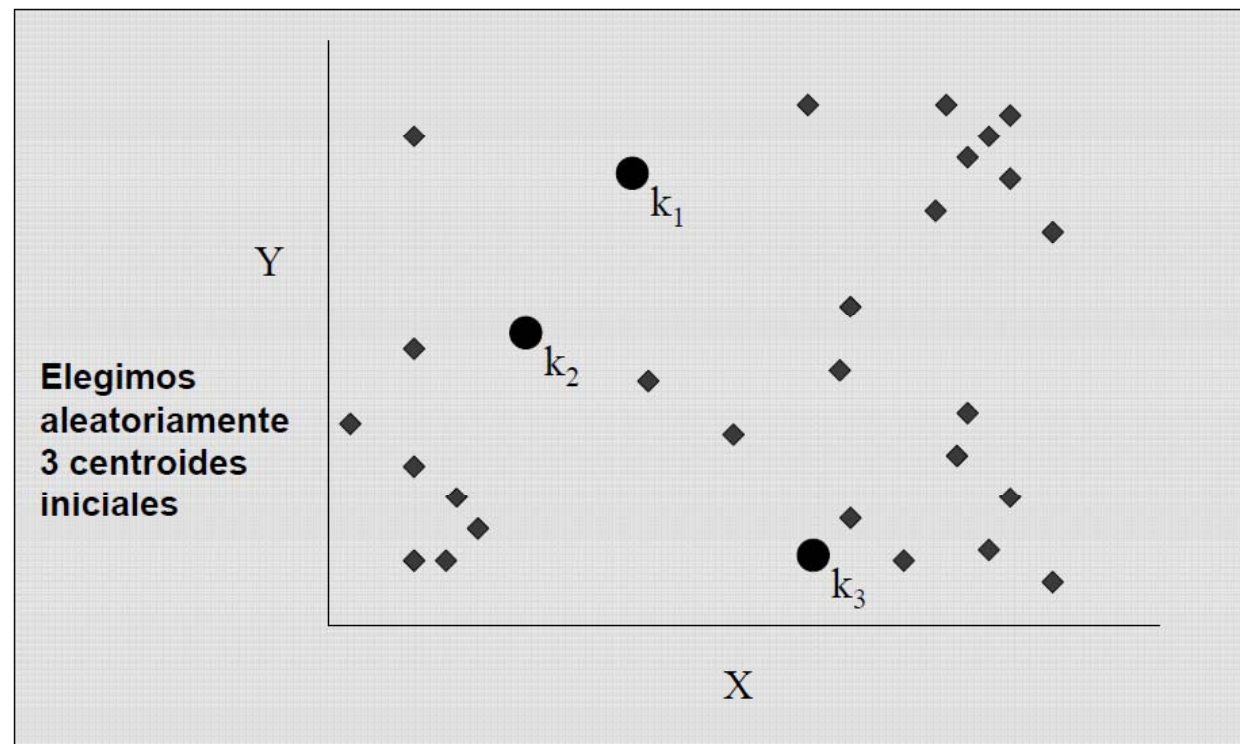
- Entradas: conjunto de ejemplos de entrenamiento, **k** grupos que se quieren obtener y medida de similitud.
 - **Paso 1:** Elegir una partición inicial en **k** grupos al azar, con un prototipo/centroide inicial para cada grupo
 - **Paso 2:** Calcular las medias μ_i de los clusters (centroides)
 - **Paso 3:** Seleccionar secuencialmente un punto **x** del conjunto, y si corresponde, reasignarlo al cluster que minimiza $d(x, \mu_i)$
 - **Paso 4:** Si no hay reasignaciones en todo el conjunto terminar; sino volver al Paso 2.
- Se realizan diferentes particiones iniciales con diferentes prototipos/centroides y se comprueba cuál partición minimiza el criterio de agrupamiento.

3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

k-medias: no jerárquico y no paramétrico

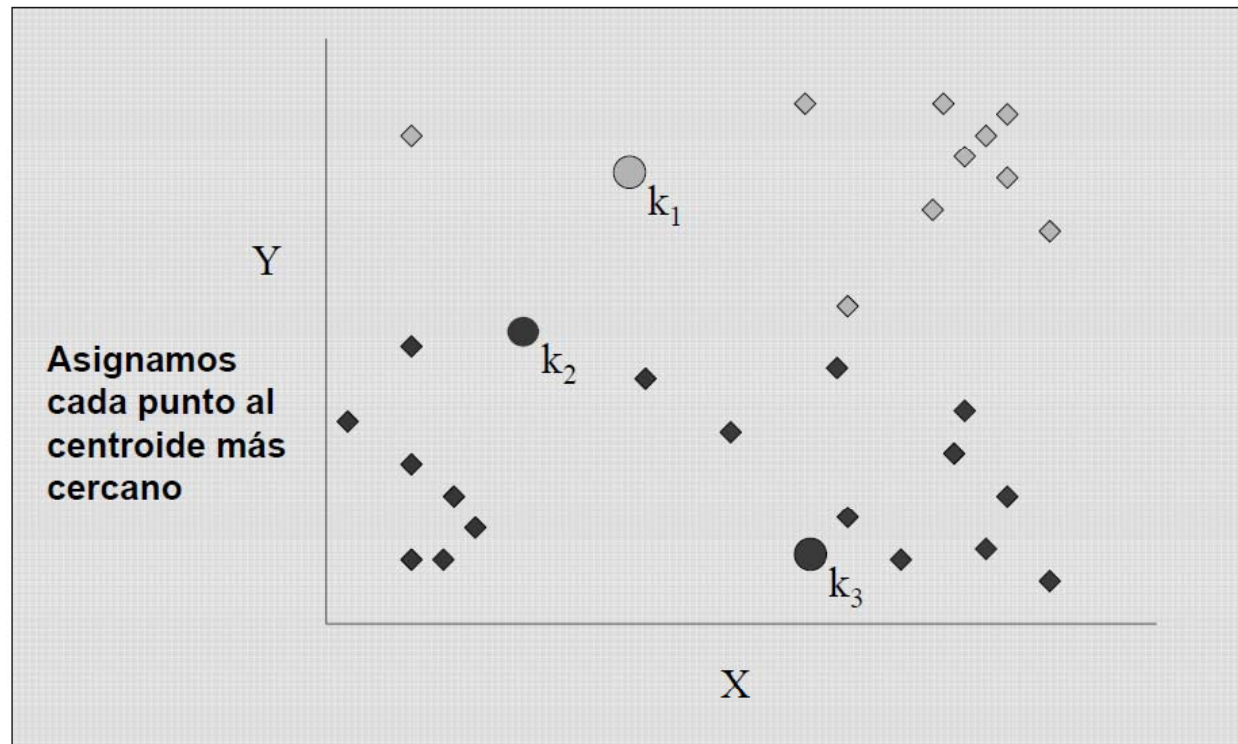


3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN k-medias: no jerárquico y no paramétrico



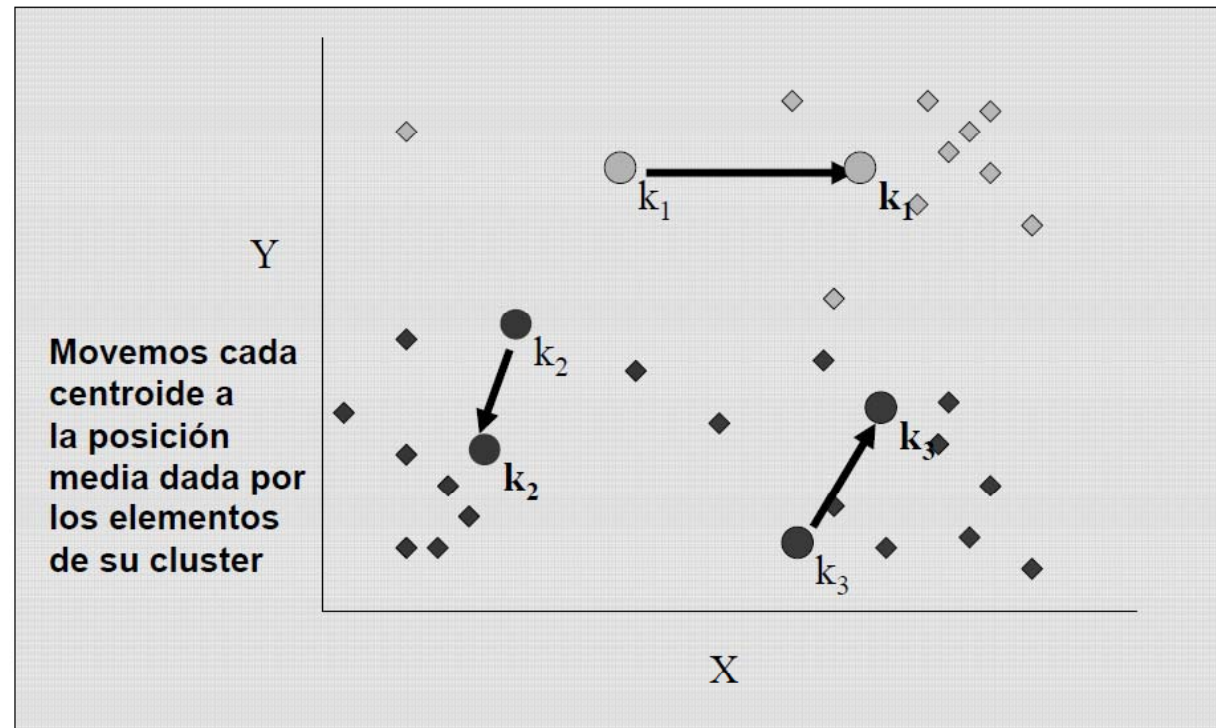
3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

k-medias: no jerárquico y no paramétrico



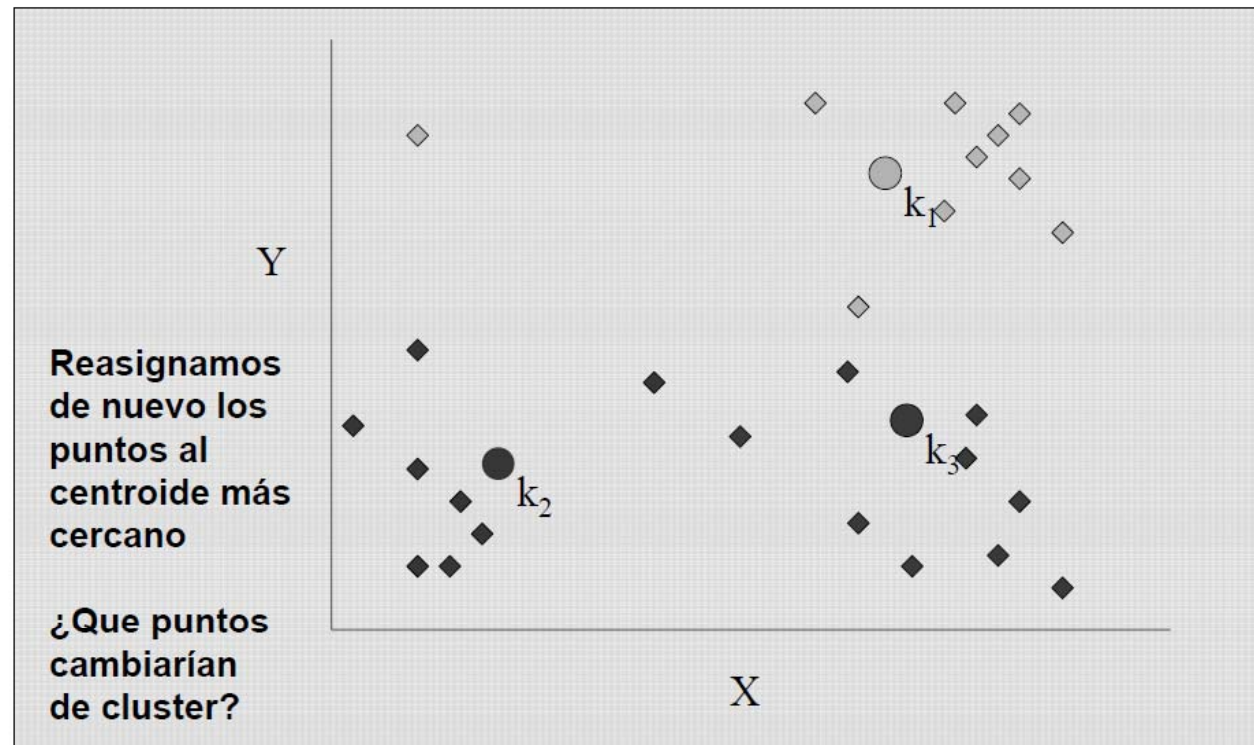
3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

k-medias: no jerárquico y no paramétrico



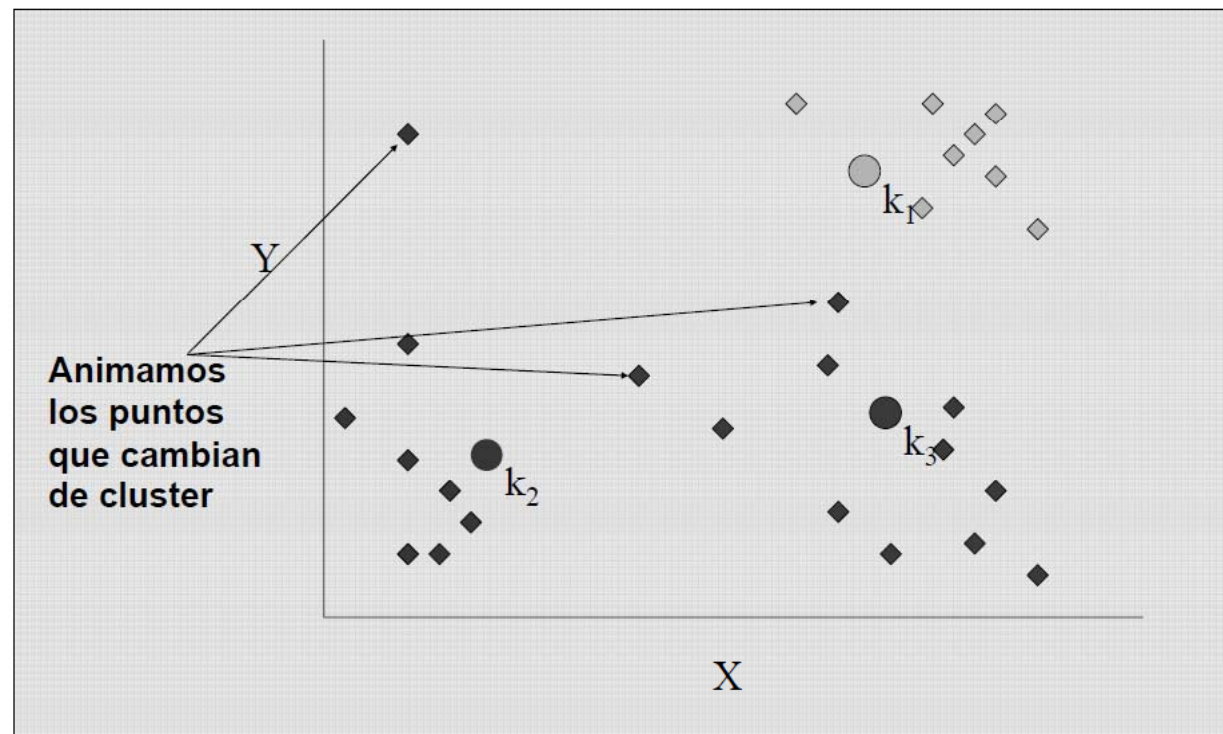
3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

k-medias: no jerárquico y no paramétrico



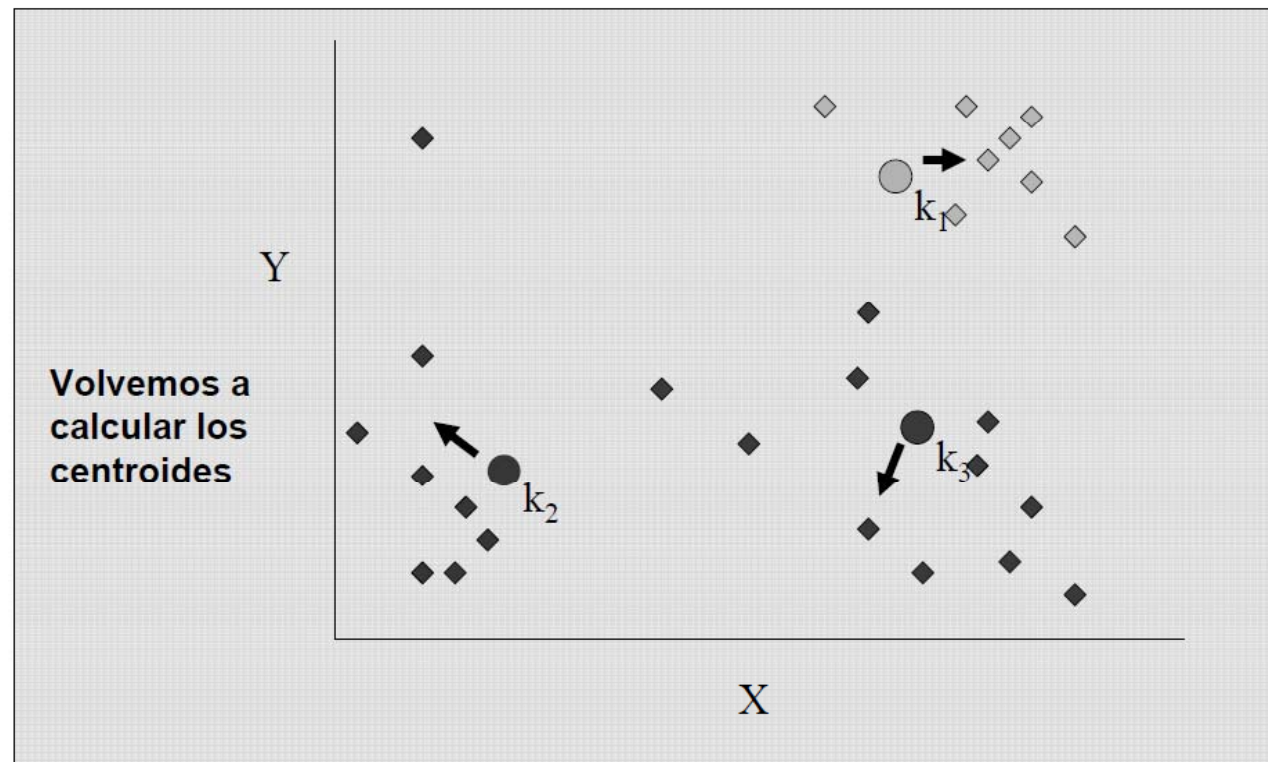
3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

k-medias: no jerárquico y no paramétrico



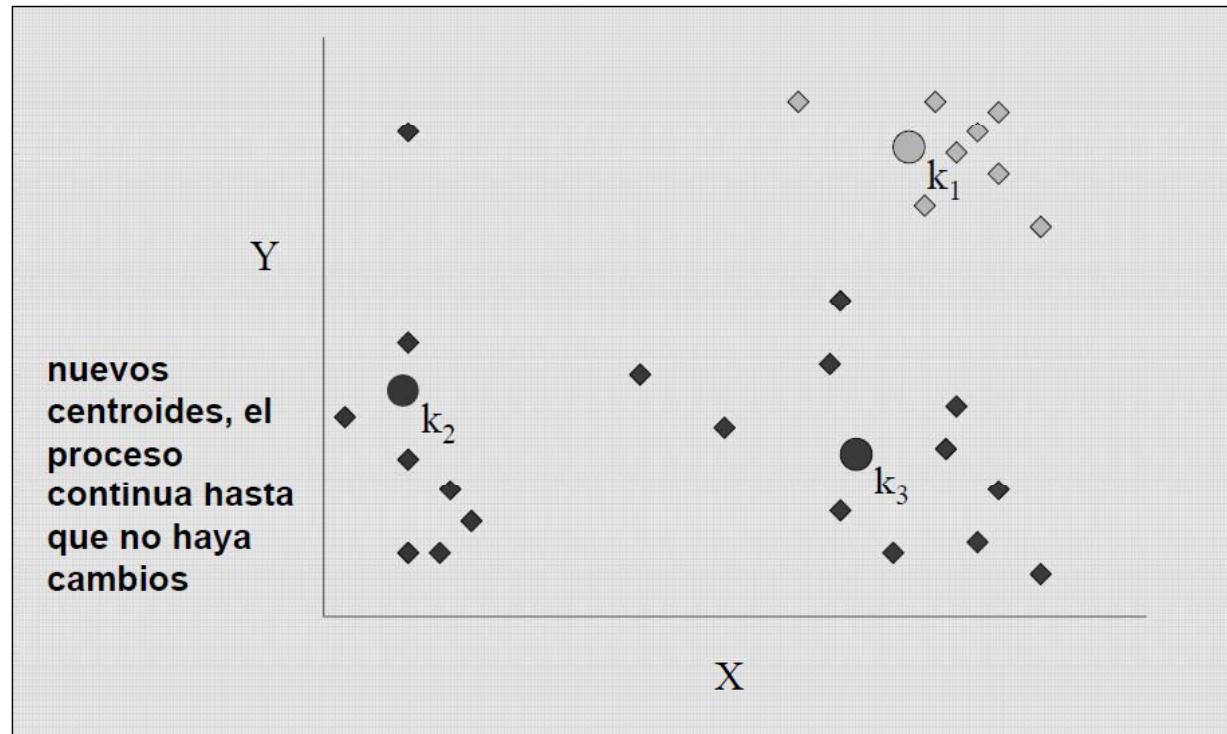
3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

k-medias: no jerárquico y no paramétrico



3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

k-medias: no jerárquico y no paramétrico



3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

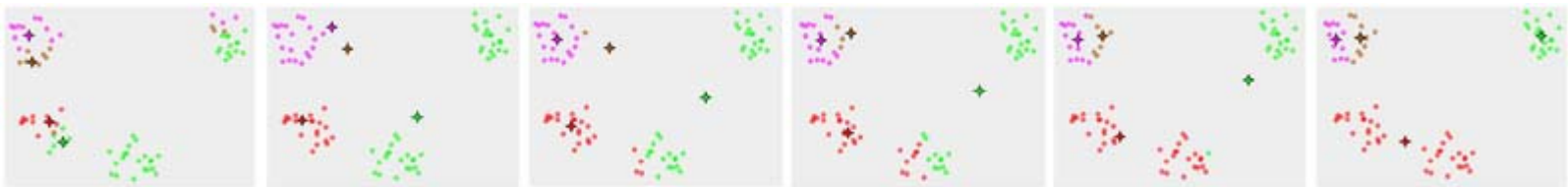
k-medias: no jerárquico y no paramétrico

- **Ventajas:**

- Simple
- Fácil de entender
- Los elementos se asignan de forma automática a clusters

- **Inconvenientes:**

- Es necesario indicar de modo heurístico el número (k) de clusters de antemano
 - Si k se elige muy pequeño, se agruparían grupos distintos
 - Si k se elige muy grande, hay centros que se quedarían “huérfanos”
- Afectado por la posición inicial de los centroides
- Muy sensible a casos extremos



- Repetir con diferentes centroides aleatorios y ver cuál minimiza el criterio de agrupamiento.

3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

K-medianas: no jerárquico y no paramétrico

- Se puede utilizar la mediana en lugar de la media:
 - Media de 1, 3, 5, 7, 9 es 5
 - Media de 1, 3, 5, 7, 9, 1009 es 205
 - Mediana de 1, 3, 5, 7, 9, 1009 es 5 y 7 (6)
- Ventaja de la mediana: no se ve afectada por casos extremos.
- También se pueden utilizar las modas (k-modes) para agrupar objetos categóricos.

3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Dynamical Clustering Algorithms

- El k-means es un caso particular de una familia de algoritmos
 - Dynamical Clustering Algorithms
- Cada grupo S_i se representa por un núcleo $K_i(\mathbf{X}_i)$: dicho núcleo puede ser una función paramétrica, un conjunto de puntos u otro modelo.
- Se define una medida de similitud $\Delta(\mathbf{x}, \mathbf{K}_i)$ y el criterio de agrupamiento

$$J = \sum_{i=1}^k \sum_{j=1}^{N_i} \Delta(\mathbf{x}_{ji}, K_i) \quad \mathbf{x}_{ji} \in \mathbf{X}_i$$

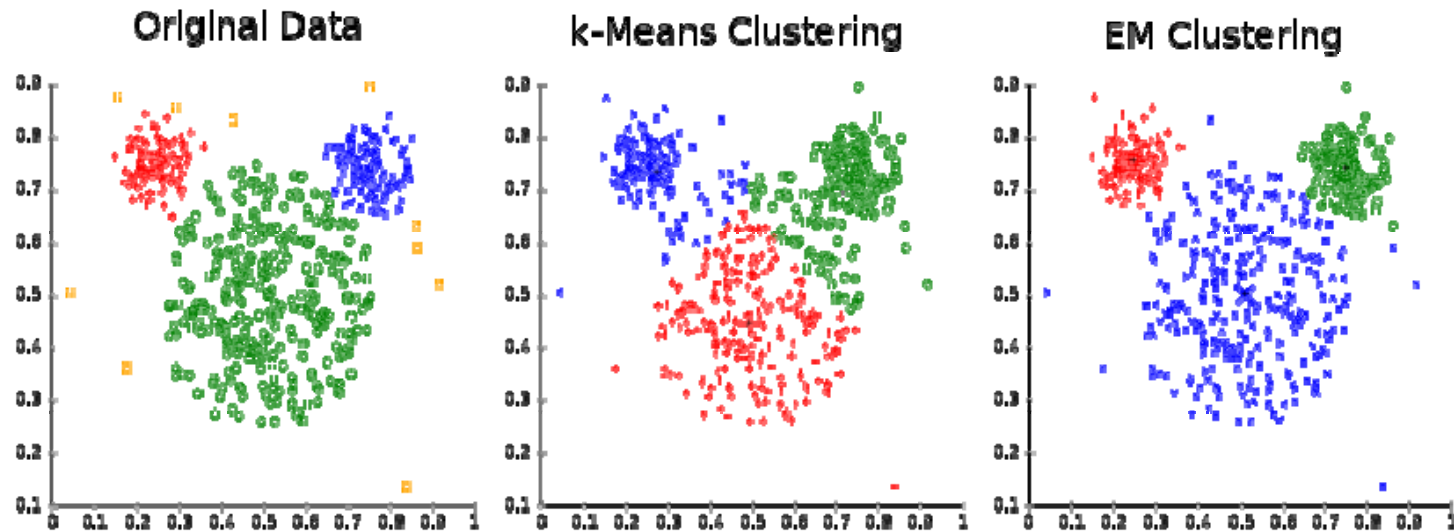
3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Dynamical Clustering Algorithms

- Por analogía, el algoritmo de agrupamiento dinámico genérico sería:
 - **Paso 1:** Elegir una partición inicial en k grupos al azar
 - **Paso 2:** Calcular el modelo $K_i(X_i)$ para cada cluster
 - **Paso 3:** Seleccionar uno a uno cada punto x del conjunto y , si corresponde, reasignarlo al cluster con mayor afinidad
 - **Paso 4:** Si no hay reasignaciones terminar el algoritmo; sino volver al Paso 2.
- El "paso de asignación" (en k-means, asociar cada punto a su centroide más cercano, según la media) también se conoce como **paso expectativa**, la "etapa de actualización" (calcular los nuevos centroides), como **paso maximización**, por lo que este algoritmo puede verse como una variante del algoritmo generalizado **expectation-maximization (EM)**.
 - Donde el modelo de representación de un cluster $K_i(X_i)$ es el modelo gaussiano.
 - Da una medida de la probabilidad de que un vector de datos pertenezca al cluster, se usa para recuperar las componentes de una mezcla de distribuciones normales.

3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Comparación algoritmos no jerárquicos

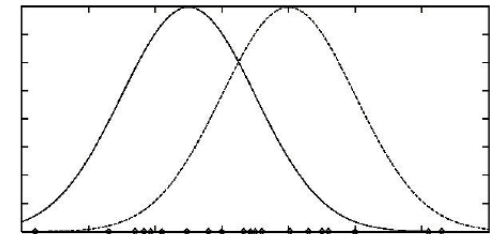


- La tendencia del k-means a producir grupos con tamaños parecidos lleva a obtener malos resultados, depende del conjunto de datos para satisfacer las necesidades del algoritmo.
- Mientras que Expectation-maximization (EM) se beneficia de la **distribución gaussiana** presente en el conjunto de datos.
- Los modelos gaussianos usados por el algoritmo EM son más flexibles ya que controlan varianzas y covarianzas.
- EM crea grupos con tamaño variable más fácilmente que k-means.

3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Algoritmo EM: no jerárquico, paramétrico y probabilístico

- Expectation Maximization, Maximum Likelihood Estimate (Dempster et al. 1977)
 - Es la base de muchos algoritmos: Autoclass, etc.
 - Se utiliza cuando existen variables ocultas o latentes
 - Se adapta al clustering: $\langle x, \text{¿grupo?} \rangle$
 - Referencia:
 - Comparación algoritmos (K-means, EM).
<http://www.sc.ehu.es/jiwdocoj/remis/docs/GarreAdis05.pdf>
- Idea intuitiva, consta de dos pasos:
 - Paso de **Estimación**: dada la hipótesis actual h (medias de las distribuciones – como centroides en k-means), estimamos las variables ocultas (probabilidad de pertenencia a grupos – la suma de todas las probabilidades será 1)
 - Paso de **Maximización**: una vez estimadas las variables ocultas, generamos una nueva hipótesis h' (movemos las medias – centroides en k-means)
 - Se repiten los pasos EM hasta que no cambia la hipótesis.
 - Aunque EM garantiza convergencia, esta puede ser a un máximo local, por lo que se recomienda repetir el proceso varias veces.



3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Algoritmo EM: no jerárquico, paramétrico y probabilístico

- Permite estimar, dado un conjunto de datos generado con c distribuciones gaussianas, las **medias** de las distribuciones: $\langle \mu_1, \dots, \mu_c \rangle$
 - Busca los parámetros que maximizan la probabilidad de pertenecer a un cluster.
- El problema es que no sabemos de qué distribución viene cada dato y no conocemos los parámetros de las distribuciones.
- El algoritmo EM (Expectation Maximization) realiza hasta converger:
 - El cálculo de las probabilidades de pertenecer a las clases o los valores esperados de las clases - es la parte de **expectation**.
 - El cálculo de los valores de los parámetros de las distribuciones (núcleo o centroide de las clases) - es **maximization**, maximizar la verosimilitud de las distribuciones dados los datos.
 - El ajuste de los parámetros del modelo requiere una medida de su bondad, es decir, cómo de bien encajan los datos sobre la distribución que los representa. Este valor de bondad se conoce como el likelihood de los datos.
 - Se trataría entonces de estimar los parámetros buscados, maximizando este likelihood (este criterio se conoce como ML-Maximum Likelihood).
 - Se puede empezar adivinando (dando valores iniciales) a los parámetros de las distribuciones – centroides o adivinando las probabilidades de que un objeto pertenezca a una clase.

3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Clustering o Agrupación no supervisada

- Los algoritmos de clustering **no supervisados** pueden ser según su salida:
 - **Jerárquicos**: jerarquía de grupos en los que unos incluyen a otros. Un grupo raíz contiene a todos los demás.
 - Ej.: Estrategia Aglomerativa; COBWEB (probabilístico)
 - **No jerárquicos o de Partición.**
 - Ej.: k-means, EM

3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

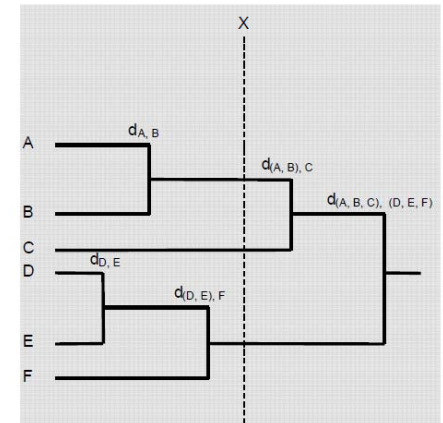
Cluster jerárquico: diferentes aproximaciones

- Top-down (disociativos/divisivos):
 - Empieza con un gran cluster.
 - Recursivamente parte en dos cada cluster.
- Bottom-up (aglomerativos).
 - Empieza con un cluster por dato.
 - Iterativamente encuentra 2 clusters para unir.
 - Los clusters se funden al encontrar pares con máxima similitud.
- Aproximaciones Probabilísticas:
 - Define un modelo probabilístico con un árbol latente y aprende la estructura del árbol mediante máxima-verosimilitud u otras técnicas.
- La aproximación dominante es bottom-up.

3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Técnicas Aglomerativas: jerárquica y no paramétrica

- Se generan unos árboles: **dendrogramas**
- Búsqueda bottom-up:
 - Inicialmente, cada nodo representa un ejemplo.
 - Se repite N-1 veces (siendo N el número de ejemplos):
 - Se calcula la similitud/proximidad entre todo par de ejemplos
 - Se agrupan los dos más cercanos
 - Se sustituyen los dos nodos por su representante (centroide o prototipo).
- No sólo se generan grupos, sino también una jerarquía entre ellos.



3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Técnicas Aglomerativas: jerárquica y no paramétrica

- Descripción del algoritmo general:
 - En cada iteración busca los 2 nodos más parecidos, de acuerdo al criterio de similitud, y crea un nuevo nodo más abstracto/general que describe a esos 2 nodos.
 - A partir de ese momento, el algoritmo no considera sus nodos inferiores para la creación de nodos más abstractos.
 - Continúa hasta llegar un momento en el que hay un nodo superior, el nodo raíz de la jerarquía.
- Consideraciones:
 - Elegir **la medida utilizada de proximidad/similitud** entre clusters.
 - La más utilizada es la distancia euclídea.
 - Es importante conocer el objetivo final del clustering.
 - Por ejemplo, puede interesar obtener todo el dendrograma (para separar grupos de animales o plantas) o tan sólo una posible partición de la jerarquía para obtener un n° concreto de clusters para un posterior análisis.

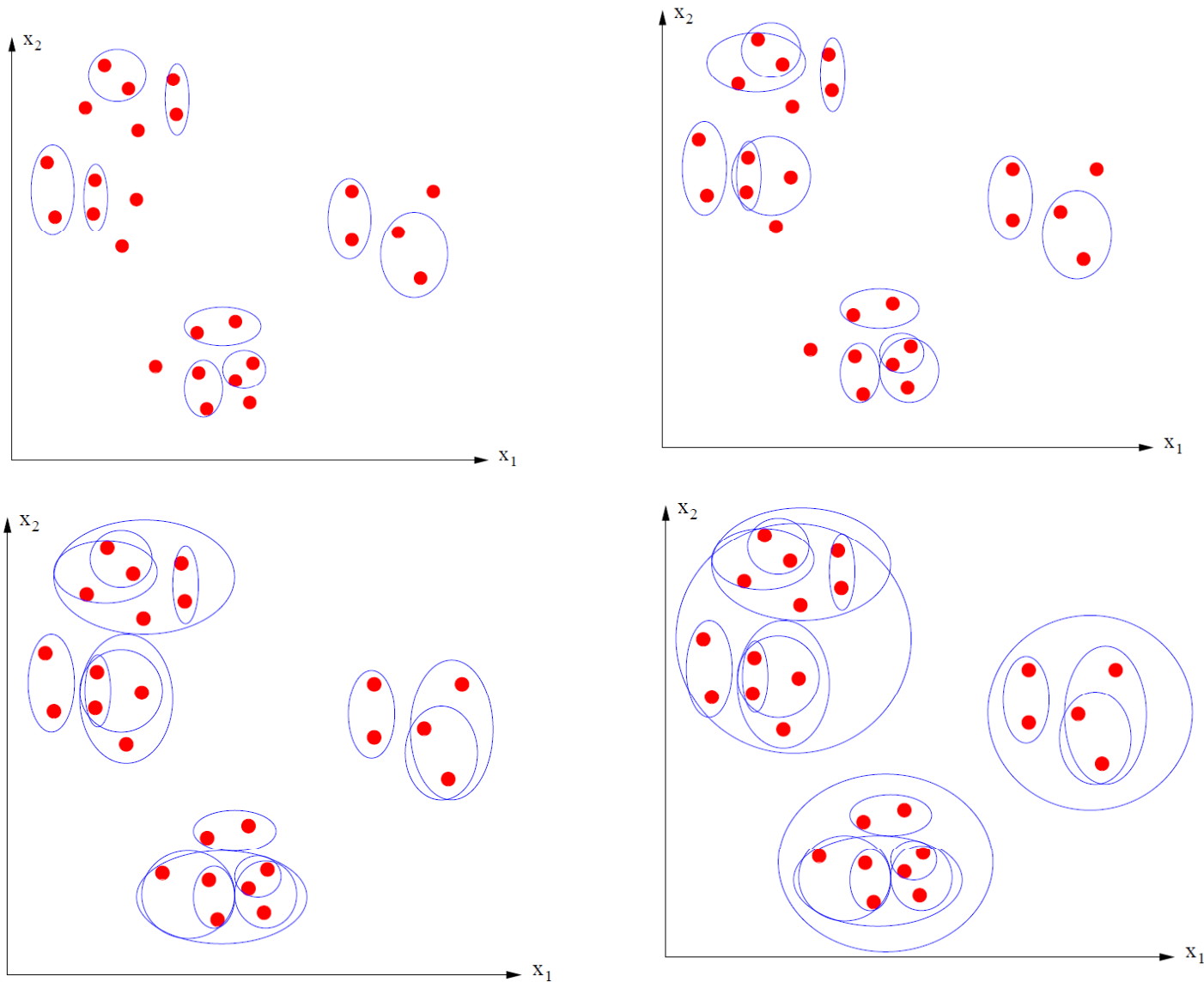
3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Técnicas Aglomerativas: jerárquica y no paramétrica

- Consideraciones (cont.):
 - Si algún valor de un atributo del ejemplo o patrón tiene un rango de valores muy grande y diferente a otros atributos, se suele normalizar (Ej. Restando la media y dividiendo por la desviación típica).
 - Una vez se juntan dos nodos, hay que decidir cómo calcular los valores de los atributos de ese nuevo nodo que los representa: calcular la media de cada atributo de los dos nodos; distancias mínimas; distancias máximas, etc.
 - Si los valores no son numéricos, se pueden convertir a coeficientes que tengan en cuenta los diferentes valores (Ej. alto = 1, medio= 0.5, bajo = 0).
 - Tener cuidado con la pérdida de información de las conversiones.
 - Una vez realizada una jerarquía, si se quiere incorporar un dato nuevo, se va comparando desde el nodo raíz y bajando sucesivamente por el resto de nodos según el detalle requerido.

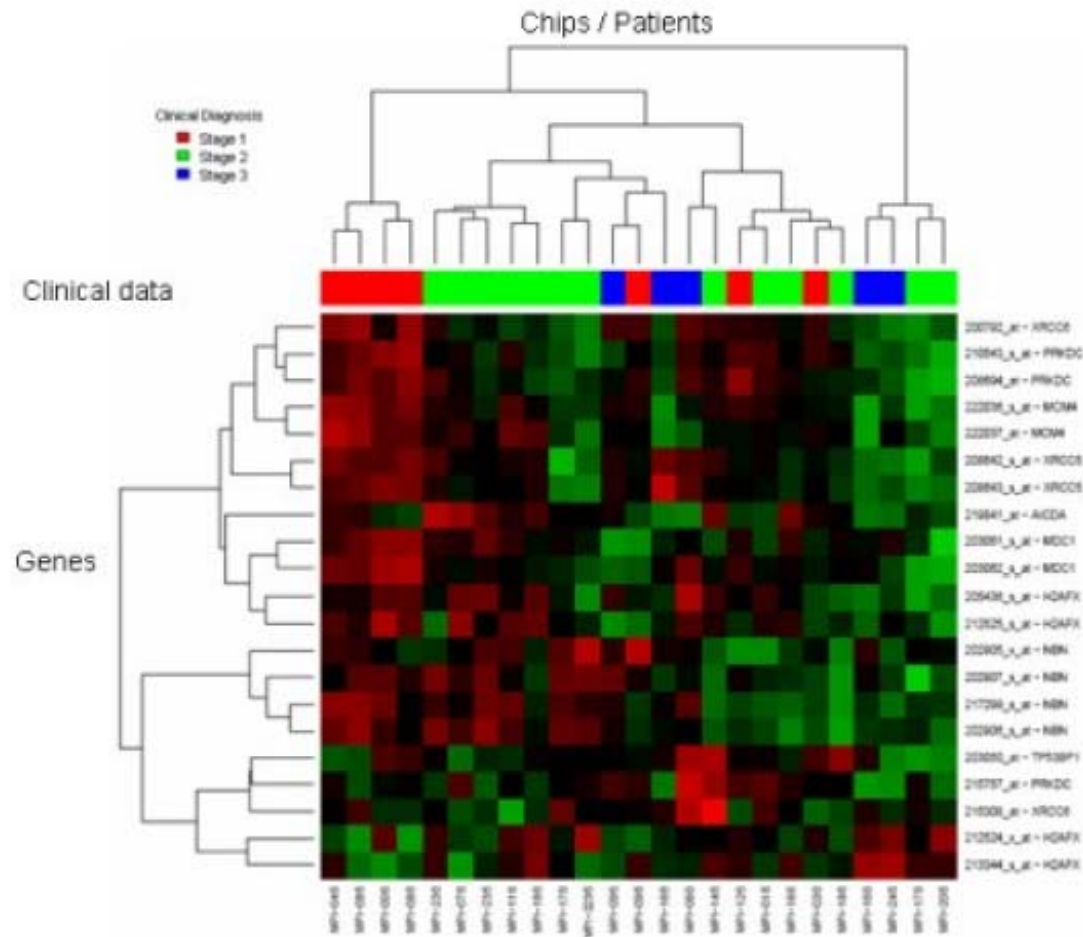
3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Técnicas Aglomerativas



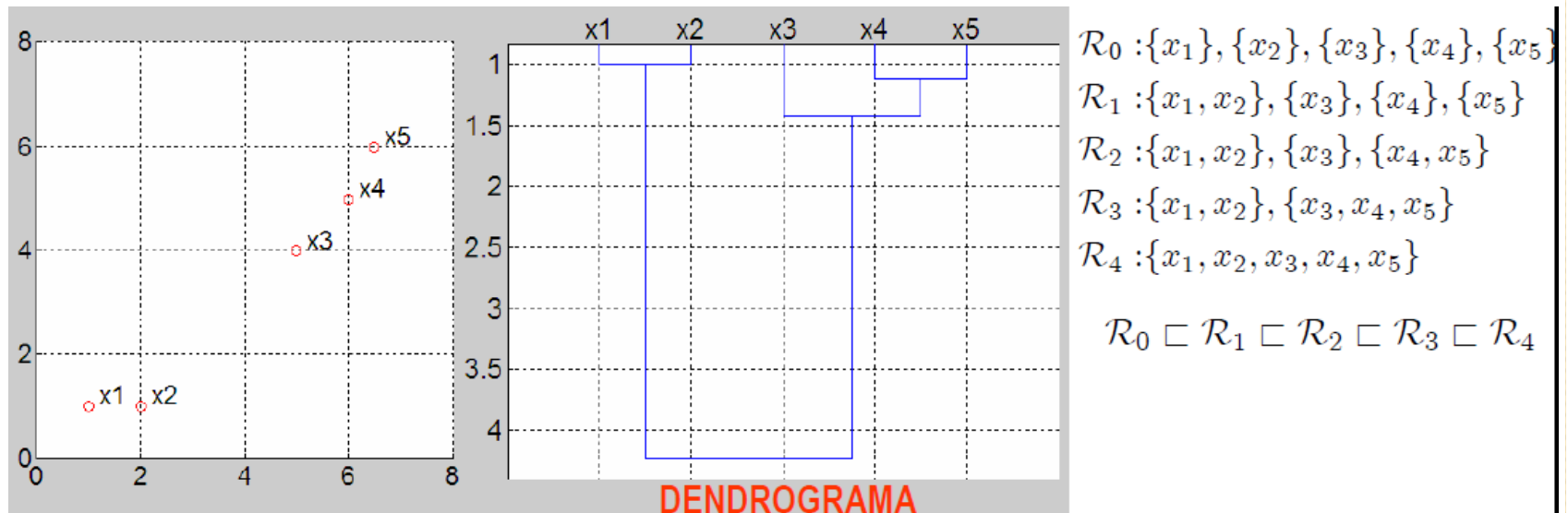
3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Técnicas Aglomerativas



3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Técnicas Aglomerativas

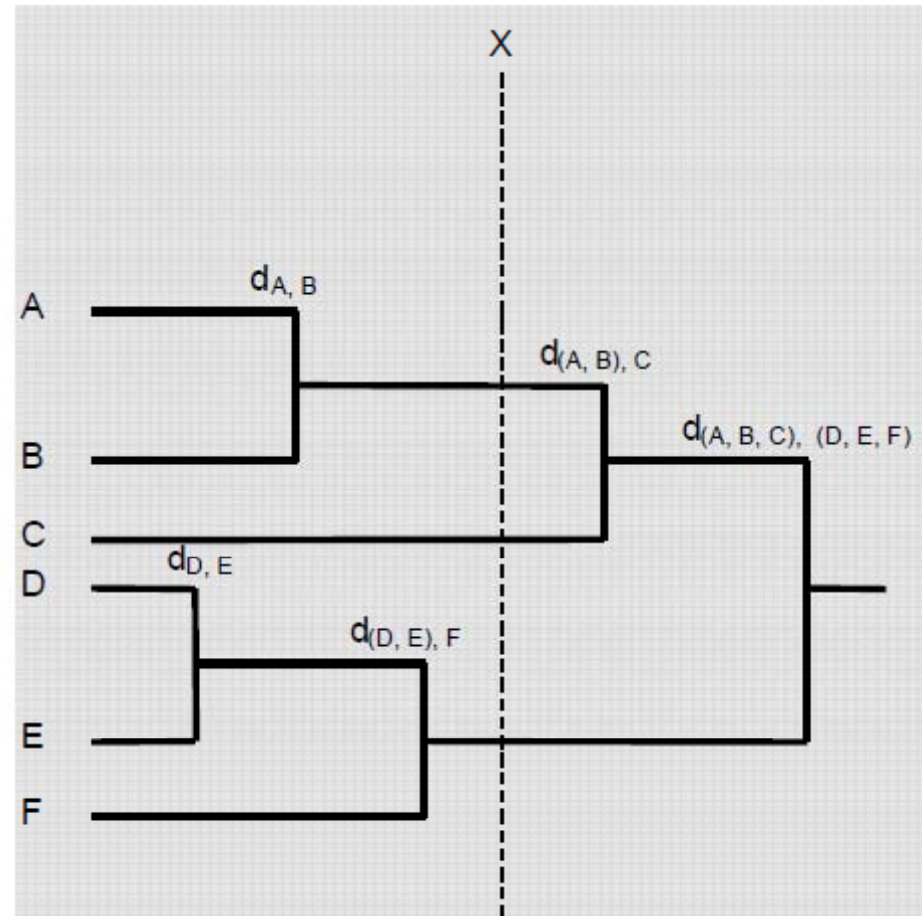


- N ejemplares \Rightarrow jerarquía de N niveles $N-1$ pasos de ejecución
- Tipos de algoritmos de agrupamiento jerárquico:
 - **Aglomerativos:** $\{x_1\}, \{x_2\}, \dots, \{x_N\} \rightarrow \{x_1, x_2, \dots, x_N\}$
 - **Divisivos:** $\{x_1, x_2, \dots, x_N\} \rightarrow \{x_1\}, \{x_2\}, \dots, \{x_N\}$

3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Técnicas Aglomerativas

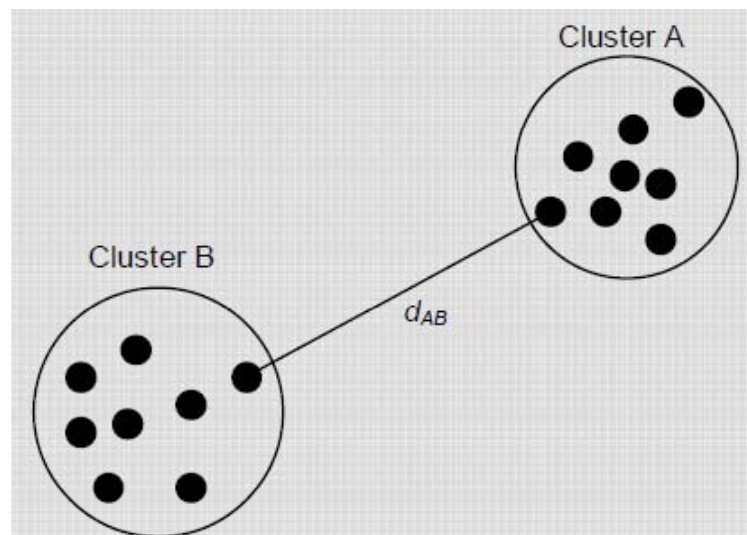
- Se forman grupos en pasos sucesivos, generando el árbol de partición o *dendrograma*.
- Si partimos el árbol en X , obtendremos los cluster cuya distancia entre cualquier par de elementos dentro del mismo cluster es menor que X .



3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Técnicas Aglomerativas

- Método de las **distancias mínimas** (single linkage)
 - Se denomina también como “vecino más cercano”
 - Es uno de los métodos más sencillos, y más utilizados.
 - Consiste en definir la distancia entre los nuevos clusters que se van formando como la distancia existente entre sus elementos más próximos.



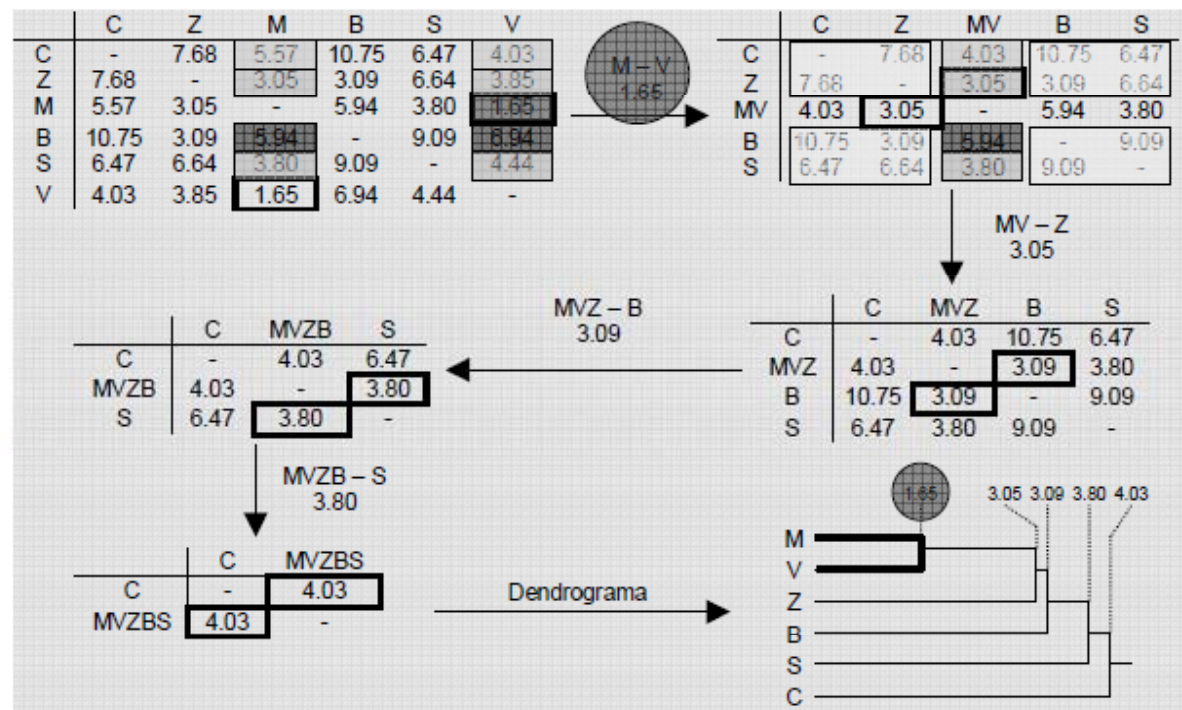
3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Técnicas Aglomerativas: algoritmo SAHN

(Sequential, Agglomerative, Hierarchical, Nonoverlapping)

1. Fijamos el número de clustering a cero: $m = 0$.
2. Repetimos los siguientes pasos hasta que $m = n$.
3. Encontramos el par de clusters i y j cuya distancia es mínima:
4. Incrementamos m en uno.
5. Mezclamos los clusters i y j en un solo cluster para definir el clustering m .
6. Definimos el nivel de este clustering como: $L(m) = dij$
7. Actualizamos la matriz de distancias borrando las filas y columnas que pertenecen a los clusters i y j , y añadiendo una nueva fila y una nueva columna para el cluster recién formando ij .
8. La distancia entre este nuevo cluster y los ya existentes depende del método

	A Coruña	Zaragoza	Madrid	Barcelona	Sevilla	Valladolid
A Coruña	---	7.68	5.57	10.75	6.47	4.03
Zaragoza	7.68	---	3.05	3.09	6.64	3.85
Madrid	5.57	3.05	---	5.94	3.80	1.65
Barcelona	10.75	3.09	5.94	---	9.09	6.94
Sevilla	6.47	6.64	3.80	9.09	---	4.44
Valladolid	4.03	3.85	1.65	6.94	4.44	---



METODO FUSION CLUSTERS: DISTANCIAS MINIMAS



3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Técnicas Aglomerativas: jerárquica y no paramétrica

- **Ventajas:**
 - No hay que definir a priori un número de grupos.
- **Inconvenientes:**
 - Cuando se trata de una agrupación conceptual (datos simbólicos, no numéricos, ej. jerarquía en biología) el proceso de agrupación necesita en muchas ocasiones conocimiento del humano.
 - Al igual que en k-means, si los atributos son simbólicos no es adecuada una medida de similitud de distancia.



3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Cluster jerárquico: diferentes aproximaciones

- Top-down (disociativos/divisivos)
- Bottom-up (aglomerativos)
- **Aproximaciones Probabilísticas**

3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Aprendizaje Probabilístico: clustering jerárquico y conceptual

- Cuando los valores de los atributos de los datos son simbólicos/conceptuales y no numéricos, es útil disponer de sistemas capaces de trabajar con dichos datos.
- Existen sistemas como COBWEB que utilizan el concepto de probabilidad para establecer en forma de árbol los grupos en los que dividir los ejemplos en el aprendizaje no supervisado.
 - Se organizan los grupos en una **jerarquía de conceptos de acuerdo a criterios probabilísticos** inspirados en algunos resultados de psicología cognitiva (Fisher, 1987; Gluck and Corter, 1985)
 - Para cada dominio y problema, existe un nivel de jerarquía mental de conceptos en los que se clasifica cada ejemplo.
 - Ej.: mostrando un lápiz rojo, cualquier persona dirá que pertenece al concepto de lápiz, pero no al de material de oficina.
 - COBWEB trata **incrementalmente** los ejemplos, encuadrando cada uno:
 - Dentro de un grupo ya creado en el árbol.
 - Creando uno nuevo.
 - Borrando algún grupo .
 - O cambiando alguno de nivel de detalle en caso de que la heurística aconseje hacerlo.

3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Aprendizaje Probabilístico: clustering jerárquico y conceptual

- HEURÍSTICA: Considerando el n° de ejemplos que pertenecen a un grupo en función de las coincidencias de sus atributos entre sí o las diferencias, se van calculando probabilidades y distribuyendo los ejemplos según éstas, recorriendo el árbol en sentido descendente.
 - Se deseará que los ejemplos que pertenezcan a distintos grupos tengan el menor n° de atributos con igual valor y los que pertenezcan a los mismos grupos tengan el mayor n° de atributos con igual valor.
 - La calidad de una partición, se medirá maximizando los parámetros de similitud intra-clase y disimilitud inter-clases.
 - Cuanto más grande es la proporción de elementos del grupo que tienen ese **atributo-valor**, ese atributo-valor es más predictivo sobre el grupo.
 - Habrá que maximizar o minimizar la probabilidad de que dado que el atributo A tenga el valor V, el ejemplo correspondiente pertenezca al grupo G.
 - COBWEB <http://ccc.inaoep.mx/~emorales/Cursos/NvoAprend/node79.html>
- Otras técnicas y variantes:
 - AutoClass, Classit, CYRUS.

3.1 APRENDIZAJE NO SUPERVISADO: AGRUPACIÓN

Validación e interpretación del clustering

- Las técnicas de clustering se emplean para descubrir estructuras en los datos, sin embargo, su modo de operar es de “imposición de estructura”.
 - Es decir, los algoritmos siempre encontrarán una estructura en grupos aunque ésta realmente no sea siempre la más adecuada.
- Por ello, es importante dilucidar si los grupos descubiertos por el análisis son reales o son impuestos por el método utilizado.
 - Usar medidas estadísticas de la calidad de los clusters.
- También es importante conocer el objetivo final del clustering.
 - Se puede desear sólo una posible partición de la jerarquía para obtener el nº de clusters deseados para un posterior análisis.
 - El clustering es a veces subjetivo, no existe la respuesta correcta, debería ser evaluado basándose en cuánto ha ayudado a alcanzar un objetivo.

TEMA 3

- 3.1 Aprendizaje no supervisado: agrupación
- **3.2 Redes de neuronas no supervisadas**

3.2 REDES DE NEURONAS NO SUPERVISADAS

- Se han visto SOM, GCS, GNG, Hopfield
- ART **Adaptive Resonance Theory** (Teoría de la Resonancia Adaptativa)

3.2 REDES DE NEURONAS NO SUPERVISADAS

REDES DE NEURONAS NO SUPERVISADAS. AUTOORGANIZATIVAS

- La **autoorganización** en Sistemas Conexionistas: consiste en la modificación repetida de los pesos de las conexiones en respuesta a modos de activación y siguiendo unas reglas preestablecidas, hasta el desarrollo final de la estructura o sistema.
 - No existen observadores globales, no existe un supervisor externo que determine el comportamiento para su aprendizaje, no se cuenta con la *salida deseada*.
 - Comportamiento emergente: características surgen de forma inesperada a partir de la interacción entre los componentes del sistema.
 - La red crea su propia representación de la información que recibe mediante la etapa de aprendizaje.
 - La información relevante debe de ser localizada en los propios datos de entrada (redundancia), sin redundancia sería imposible encontrar características en los datos.
 - Son capaces de modificar sus parámetros internamente, adaptándose al entorno de la mejor manera posible.

3.2 REDES DE NEURONAS NO SUPERVISADAS

REDES DE NEURONAS NO SUPERVISADAS. AUTOORGANIZATIVAS

- ▶ **Aplicaciones:** Tipos de problemas que pueden resolver, condicionará como se interprete lo que representa la salida de la red
- **Agrupamiento de patrones (clustering):** A partir de un conjunto de entrada se desea determinar si se puede dividir ese conjunto en diferentes grupos o *clusters* con un centro o patrón del *cluster*. Ej. Extraer temas de documentos.
 - Permite separar datos en grupos (a priori no sabemos cuántos existen)
 - Facilita indicar a qué grupo pertenece cada dato de entrada
 - Determinando su neurona ganadora
 - Permite caracterizar cada grupo
 - Mediante los pesos de la neurona que representa ese grupo
- **Prototipado:** Similar al anterior, en lugar de interés por los grupos, interesa obtener un prototipo del grupo al que pertenece cada patrón de entrada.
 - Usa los pesos de la ganadora para determinar ese prototipo
 - Ej. Compresión de imagen.

3.2 REDES DE NEURONAS NO SUPERVISADAS

REDES DE NEURONAS NO SUPERVISADAS. AUTOORGANIZATIVAS

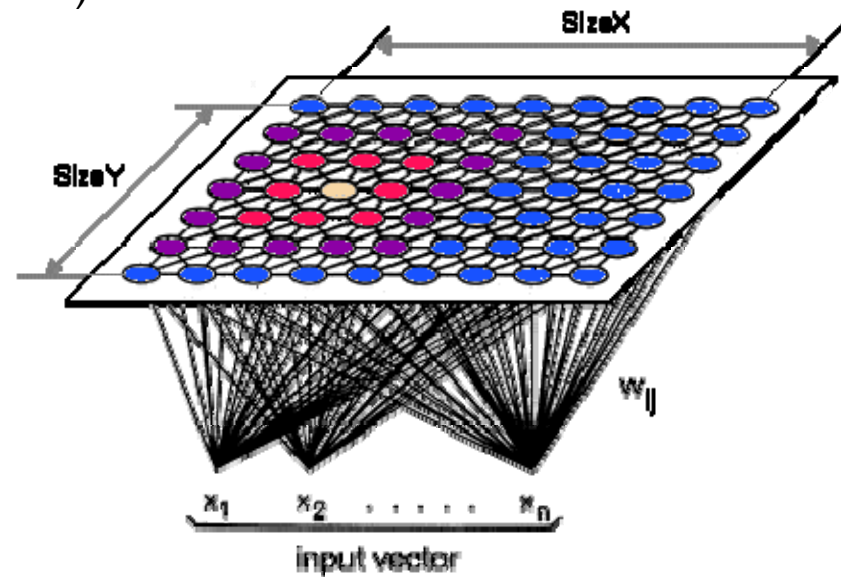
► **Aplicaciones:**

- **Análisis de componentes principales:** Se trata de detectar qué elementos/atributos del conjunto de entrada caracterizan en mayor grado ese conjunto de datos. Seleccionar las entradas de la red realmente necesarias, sin pérdida de información – supone reducción de dimensionalidad.
 - Las demás entradas podrán eliminarse sin una pérdida significativa de información.
- **Extracción y relación de características:** Se pretende organizar los vectores de entrada en un mapa topológico
 - A partir de la red entrenada, patrones parecidos producirán respuestas similares en neuronas cercanas.
 - Si existe una organización global de patrones de entrada, se verá reflejada en la salida de la red.
 - Se ubican entradas parecidas y/o relacionadas en zonas próximas de la red
- Estas categorías no son disjuntas. Ej: componentes principales para reducir la dimensionalidad y luego aplicar clustering.

3.2 REDES DE NEURONAS NO SUPERVISADAS

SOM - MAPAS AUTOORGANIZATIVOS DE KOHONEN

- El científico finlandés Teuvo Kohonen diseñó en 1982 un modelo llamado **mapa autoorganizativo de características** que consiste en una red de neuronas de 2 capas: capa de entrada y capa de competición (salida).
- ▶ Modelo de red neuronal con capacidad para formar mapas de características, simulando los mapas topológicos de los fenómenos sensoriales existentes en el cerebro, a través de una organización matricial de neuronas artificiales, que permite:
 - Conseguir un modelo simplificado de los datos de entrada.
 - Obtener un mapa que muestra gráficamente las relaciones existentes entre los datos, preservando su topología
 - Proyectar datos altamente dimensionales a un esquema de representación de baja dimensión: representar conjuntos de datos de gran número de atributos en mapas 2D.
 - Encontrar similitudes en los datos: visualmente podemos detectar de forma rápida cómo quedan agrupados patrones con valores próximos entre sí (Ej. color rojo).



3.2 REDES DE NEURONAS NO SUPERVISADAS

SOM - MAPAS AUTOORGANIZATIVOS DE KOHONEN

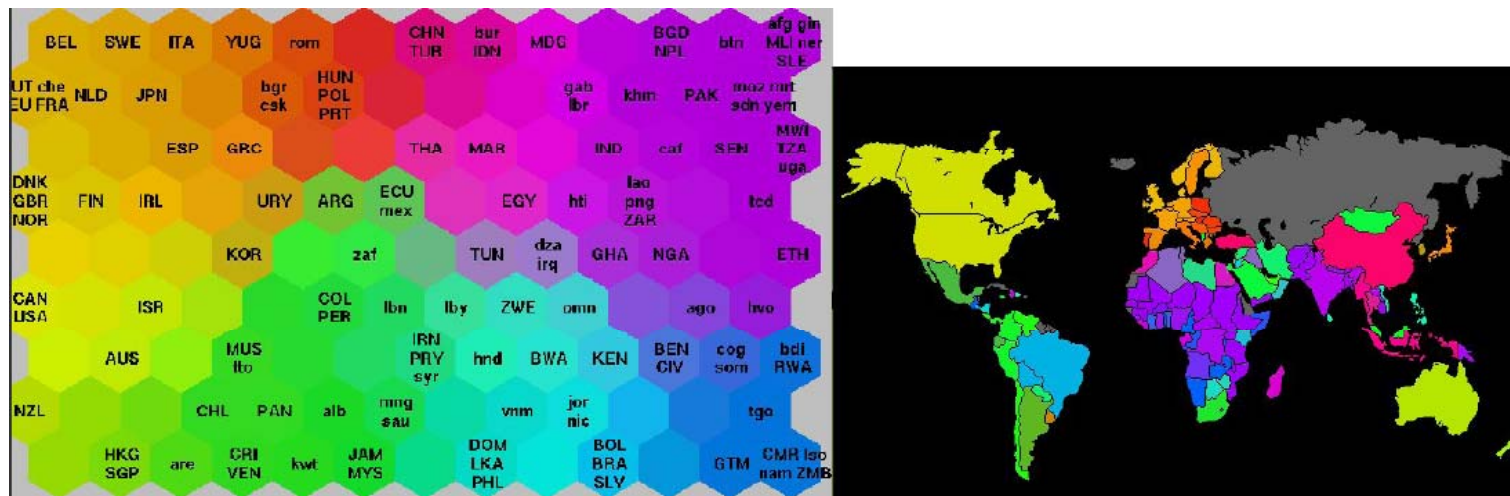
- Se trata de categorizar los patrones de entrada, agrupar los patrones similares:
 - Se persigue que patrones parecidos hagan reaccionar a las mismas neuronas.
 - Cada neurona se especializa en determinados patrones de entrada.
 - Cada grupo de patrones estará representado por un prototipo (pesos de la neurona).
 - Cada prototipo será utilizado, una vez entrenada la red, para clasificar datos nuevos y desconocidos.
- En el aprendizaje competitivo:
 - Si un patrón nuevo pertenece a una categoría reconocida previamente, entonces la inclusión de este nuevo patrón a esta categoría o clase matizará la representación de la misma.
 - Si el nuevo patrón no pertenece a ninguna de las categorías reconocidas anteriormente, entonces la estructura y los pesos de la red neuronal serán ajustados para reconocer a la nueva clase.
- 2 ideas centrales en las que se basa Kohonen para desarrollar sus estructuras usando aprendizaje autoorganizativo y competitivo: *“El proceso de adaptación de pesos y el concepto de geometría topológica de elementos de proceso”*

3.2 REDES DE NEURONAS NO SUPERVISADAS

SOM - MAPAS AUTOORGANIZATIVOS DE KOHONEN

Ejemplo Aplicación:

- ▶ Mapa de la “pobreza” mundial (Helsinki University of Technology)
<http://www.cis.hut.fi/research/som-research/worldmap.html>
 - ▶ Países descritos mediante vectores de 39 indicadores de calidad de vida (nivel educativo, sistema sanitario, etc).



- ▶ Después del entrenamiento del SOM:
 - ▶ Cada neurona de competición se “especializa” en un país.
 - ▶ Países con niveles de riqueza similares activan neuronas próximas.

3.2 REDES DE NEURONAS NO SUPERVISADAS

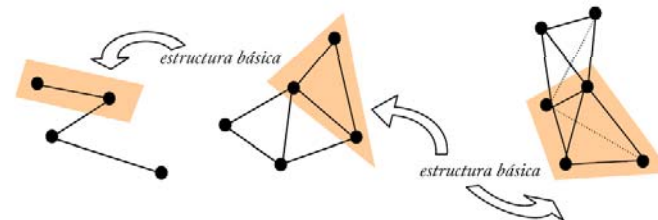
CRECIMIENTO DE REDES

- A pesar de las interesantes características del SOM de Kohonen, en la década de los años 90 se presentaron nuevos modelos de mapas autoorganizativos que pretendían solucionar los inconvenientes existentes.
- Al trabajar con SOM **hay que definir previamente la estructura de la red.**
 - Esta definición suele ser bastante compleja por la falta de información que normalmente se tiene sobre el espacio de entrada.
 - La topología estática de los SOM limita a veces su capacidad para ajustarse a los datos.

3.2 REDES DE NEURONAS NO SUPERVISADAS

CRECIMIENTO DE REDES

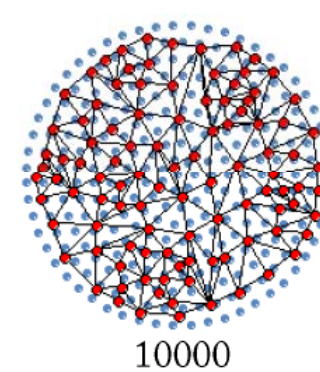
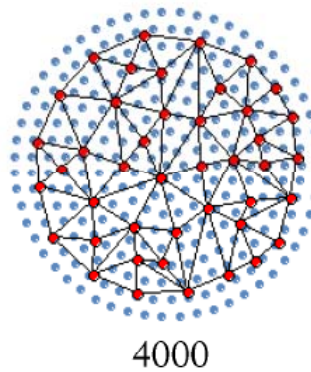
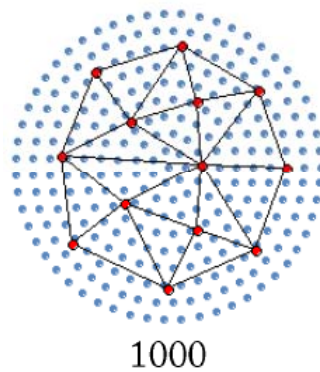
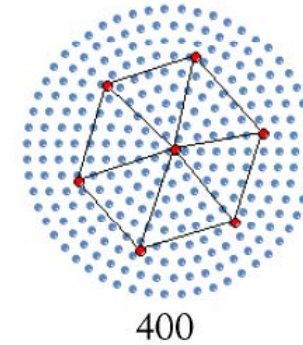
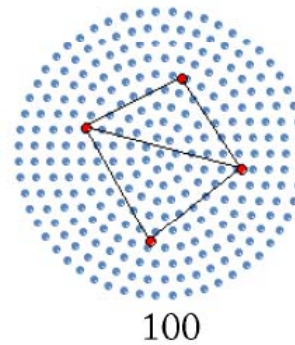
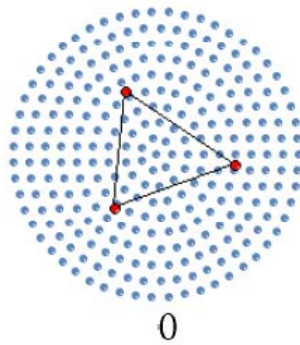
- Para solucionar este problema aparece una nueva filosofía: **Crecimiento de Redes**.
 - Se trata de iniciar el aprendizaje con un conjunto pequeño de elementos de proceso (EP) y que sea la propia red la que determine de forma autónoma su estructura, a partir de los datos de entrenamiento se generan los EP necesarios y las conexiones entre ellos.
 - Esta característica permite obtener modelos simplificados de los datos **con mejores medidas de la preservación de la topología** que las redes de Kohonen.
 - Se construyen partiendo de estructuras básicas (*hipertetraedros de dimensión k*).
 - Para llegar a la estructura final de la red se van añadiendo y borrando EP y conexiones, al tiempo que se sigue un proceso de autoorganización similar al que se produce en el modelo propuesto por Kohonen.
 - Incluso pueden existir varias mallas de vecindad separadas en la capa de salida.
 - Mantienen la capacidad de ofrecer un mapa para visualizar las relaciones existentes entre los patrones de entrada.



3.2 REDES DE NEURONAS NO SUPERVISADAS

GROWING CELL STRUCTURES (GCS)

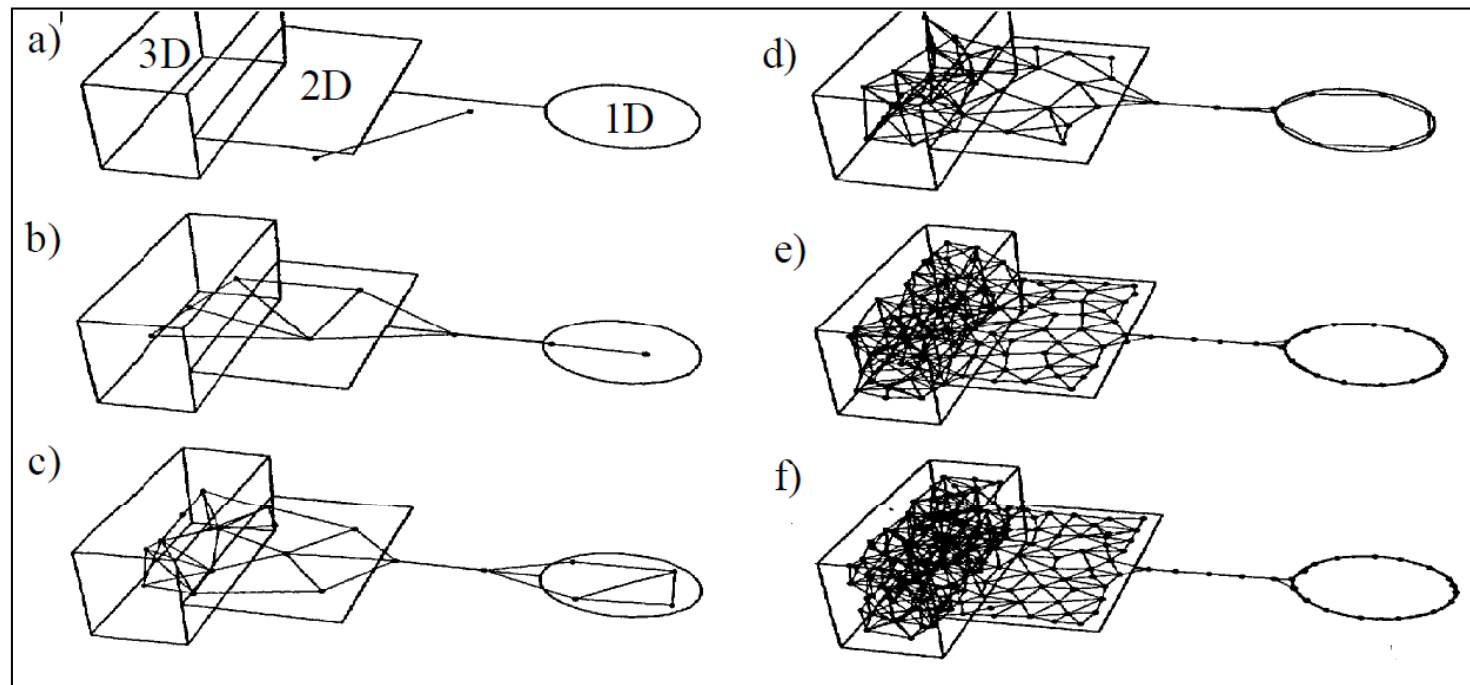
- GCS. Ejemplo:
 - Evolución del entrenamiento de una red GCS bidimensional con parámetros: $k=2$ y $\lambda=100$



3.2 REDES DE NEURONAS NO SUPERVISADAS

GROWING NEURAL GAS (GNG)

- GNG. Red autoorganizativa que se adapta a diferentes dimensiones y puede ir variando la dimensión de la estructura dimensional que emplea en función de los datos de entrada.

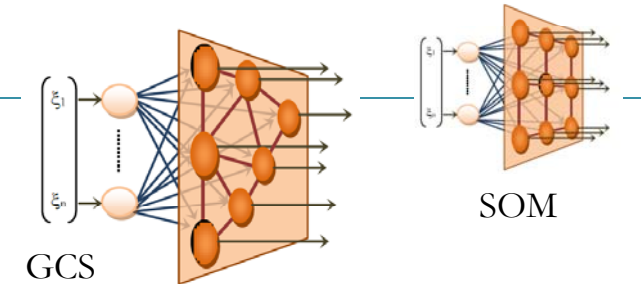


3.2 REDES DE NEURONAS NO SUPERVISADAS

CRECIMIENTO DE REDES

- Resumen características:

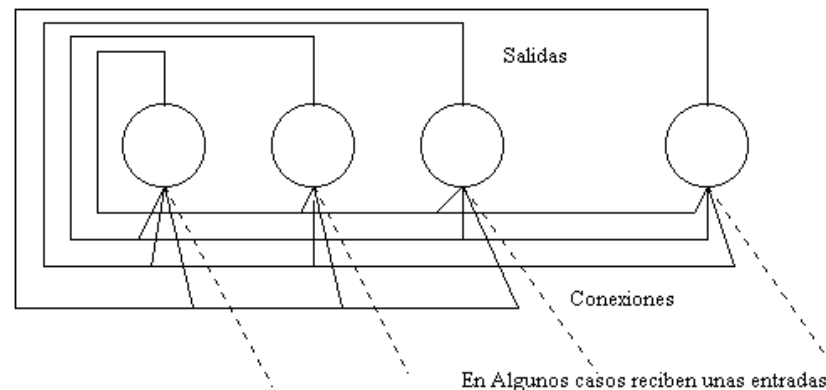
- La topología autoadaptativa de la red es altamente independiente del usuario.
- La estructura de la capa de salida de la red queda automáticamente establecida por los datos de entrada.
- Los parámetros de entrenamiento que hay que establecer son pocos y constantes, en contraste con los parámetros de área de vecindad o factor de aprendizaje dependientes del tiempo en el modelo de Kohonen.
- La capacidad de insertar y eliminar neuronas, ofrece la posibilidad de obtener mejores estimaciones de la densidad de probabilidad del espacio de entrada que los de otras redes autoorganizativas, al no posicionar neuronas en zonas del espacio de entrada con baja o nula densidad.
- Tienen suficiente capacidad para procesar espacios de datos de gran dimensión, tareas de descubrimiento de conocimiento como: aproximación de funciones, visualización (extracción de áreas o puntos en imágenes cerebrales), clustering (en datos bioinformáticos), etc.



3.2 REDES DE NEURONAS NO SUPERVISADAS

RED DE HOPFIELD (MEMORIA AUTOASOCIATIVA)

- Hopfield propone una estructura de red neuronal que viene a ser una memoria autoasociativa, **de una sola capa, totalmente conectada y recurrente**.
 - ▶ Si se usa como memoria auto-asociativa: ante la entrada de un patrón incompleto o con ruido, la red evoluciona hacia el patrón completo más cercano.



MODELO DE HOPFIELD

- Tiene tantas neuronas como entradas del vector de entrada.
- Su matriz de pesos es cuadrada y simétrica, es decir los pesos de una neurona a otra tienen el mismo valor en ambas direcciones y la diagonal son todo ceros (no hay conexión de cada neurona consigo misma).
 - Cada neurona está conectada con todas las demás, incluso consigo misma.

3.2 REDES DE NEURONAS NO SUPERVISADAS

RED DE HOPFIELD (MEMORIA AUTOASOCIATIVA)

- La información que recibe esta red debe haber sido previamente codificada y representada en forma de vector (como una configuración binaria si la red es discreta o como un conjunto de valores reales si la red es continua) con tantas componentes como neuronas.
- La operación de la red es totalmente diferente al sistema de perceptrón: en el modelo de Hopfield, la primera salida es tomada como entrada en el ciclo siguiente, produciendo una nueva salida.
- Por tanto el aprendizaje es también diferente; en este **sistema no se trata de ajustar pesos** ya que éstos se establecen y mantienen constantes desde el principio, **se trata de encontrar dichos pesos**, en función del problema.
- La red no es entrenada, realiza un proceso de aprendizaje off-line, sin embargo es posible determinar la matriz de pesos por medio de un procedimiento: contendrá los patrones almacenados.
- Una memoria autoasociativa se construye a partir de la información que se desea almacenar.

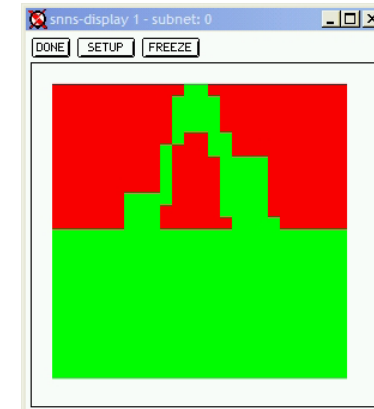
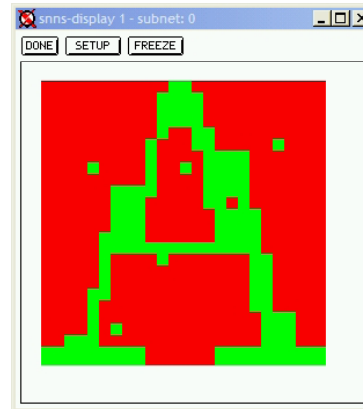
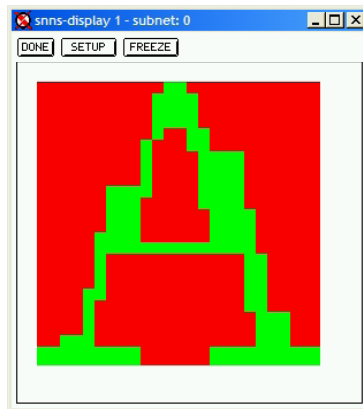
3.2 REDES DE NEURONAS NO SUPERVISADAS

RED DE HOPFIELD (MEMORIA AUTOASOCIATIVA)

EJEMPLOS DE APLICACIÓN:

Recuperación de caracteres alfanuméricos ruidosos utilizando la red de Hopfield.

- Crear inicialmente un conjunto entrenamiento formado por 5 patrones originales de dimensión 24 (matriz de 24x24), sin ruido, que serán los que la red aprenda.
- Después habrá que ir presentándole los mismos patrones, pero con ruido añadido para observar como la red es capaz de eliminar dicho ruido.
- Además se le presentarán también patrones incompletos para comprobar si la red es capaz de identificarlos:



3.2 REDES DE NEURONAS NO SUPERVISADAS

TEORÍA DE LA RESONANCIA ADAPTATIVA (ART)

- Una de las características de la memoria humana consiste en la habilidad para aprender nuevos conceptos, sin necesitar para ello olvidar otros aprendidos en el pasado.
- Objetivo:
 - Obtener esta misma capacidad en las redes neuronales.
 - Sin embargo, muchas de estas redes tienden a olvidar informaciones pasadas al tratar de enseñarles otras nuevas.
- Limitaciones que se encuentran:
 - Problemas no están bien definidos.
 - Añadir nuevos patrones de interés una vez que se ha realizado el entrenamiento con unos patrones iniciales puede suponer el olvido de lo aprendido en la primera fase.

3.2 REDES DE NEURONAS NO SUPERVISADAS

TEORÍA DE LA RESONANCIA ADAPTATIVA (ART)

- Lo que se ha intentado mostrar en esta descripción es lo que se denomina el ***Dilema de la Estabilidad-Plasticidad del Aprendizaje***.
- Este dilema plantea los siguientes interrogantes:
 - Poder aprender nuevos patrones (Plasticidad del aprendizaje).
 - Poder retener los patrones previamente aprendidos (Estabilidad del aprendizaje)
- Conseguir una red que pueda dar respuesta a uno de estos interrogantes es sencillo, pero no es sencillo diseñarla para que solucione ambos.
 - Redes conocidas, tales como el MLP, son capaces de aprender cómo han de responder ante unos patrones de entrada pero, una vez entrenados, el intentar que aprendan nuevos patrones puede suponer el "olvido" de lo aprendido previamente.

3.2 REDES DE NEURONAS NO SUPERVISADAS

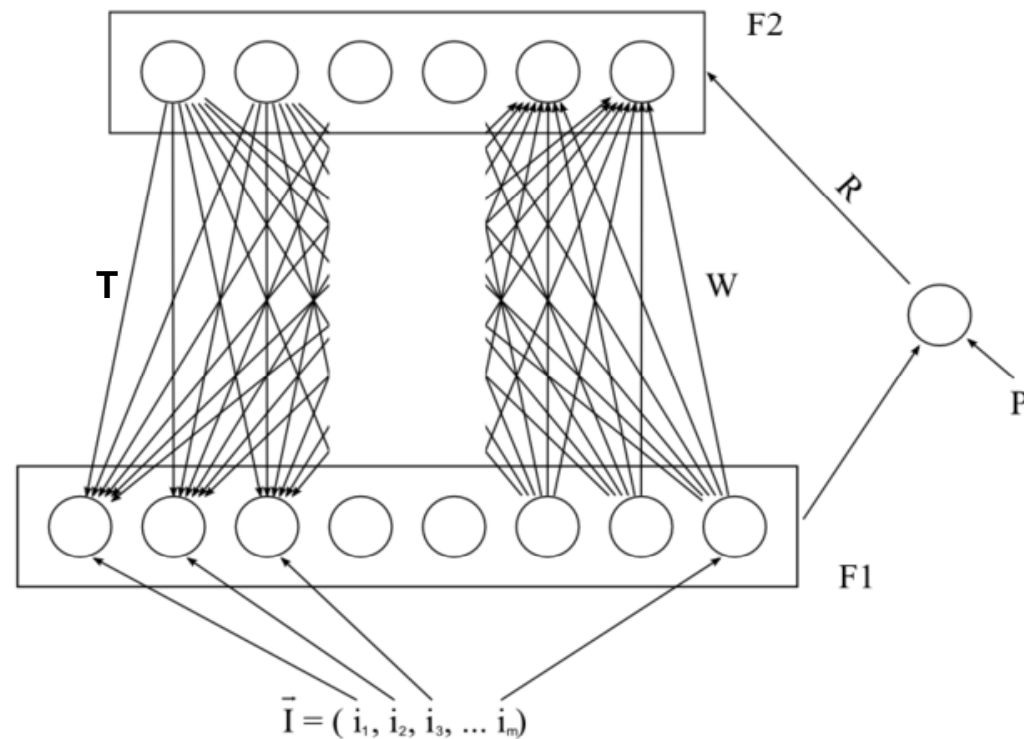
ART

- En respuesta a este dilema, Stephen Grossberg y Gail Carpenter desarrollaron la denominada Teoría de la Resonancia Adaptiva (ART).
 - Se aplica a sistemas competitivos.
 - Se pretende categorizar (clusterizar en aprendizaje Sin Supervisor).
 - El aprendizaje se produce mediante un mecanismo de realimentación creado por la competencia entre las neuronas de la capa de salida y la capa de entrada de la red.
 - A diferencia de las SOM, las ART producen automáticamente el n° de categorías óptimo a partir de los ejemplos de entrada y de un parámetro (**factor de vigilancia**), que controla la categorización de cada entrada.
 - Existen diferentes tipos de ART: primeras versiones no supervisadas ART1 (entradas binarias, 1987), ART2(entradas continuas, 1987), Fuzzy ART (1991)), versiones posteriores supervisadas (ARTMAP (1991), Fuzzy ARTMAP (1992), Distributed ARTMAP (1998), etc.

3.2 REDES DE NEURONAS NO SUPERVISADAS

ART

- Una red ART básicamente consta de 2 capas entre las que se establecen conexiones hacia adelante (pesos W) y hacia atrás (pesos T) y ciertas señales de control.



3.2 REDES DE NEURONAS NO SUPERVISADAS

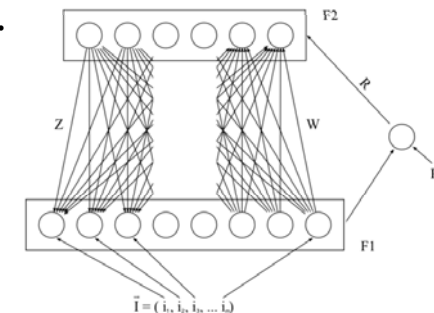
ART

- Para solucionar el dilema de la plasticidad-estabilidad, el modelo ART propone añadir un **mecanismo de realimentación** entre las neuronas competitivas de la capa de salida de la red y la capa de entrada.
- Este mecanismo facilita el aprendizaje de nueva información, sin destruir la ya almacenada.
- La **teoría de la resonancia adaptiva** se basa en la idea de hacer “resonar” la información de entrada con los representantes o prototipos (vector de pesos) de las categorías que reconoce la red.
 - Si entra en resonancia con alguno, es suficientemente similar (*cumple con vigilancia*), la red considera que pertenece a dicha categoría y únicamente **realiza una pequeña adaptación del prototipo almacenado**, representante de la categoría, para que incorpore algunas características del dato presentado (*acercamiento del centroide del cluster a la entrada*).
 - Cuando “no resuena” con ninguno, no se parece a ninguno de los prototipos existentes recordados por la red hasta ese momento (*no cumple con vigilancia*), la red se encarga de **crear una nueva categoría** con el patrón de entrada como prototipo de la misma.

3.2 REDES DE NEURONAS NO SUPERVISADAS

ART

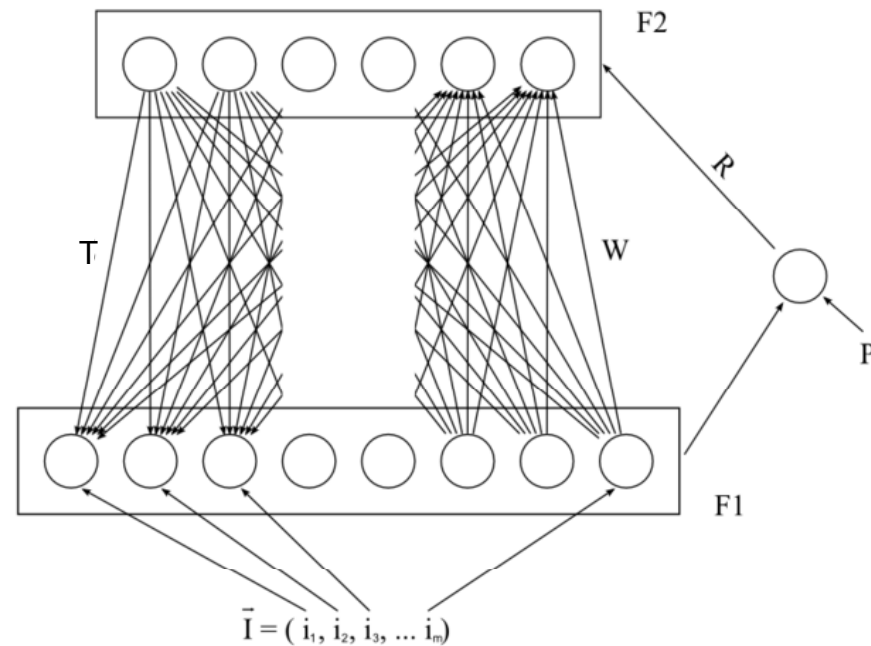
- **Capa de entrada (F1):** datos de entrada pasan a ser los valores de sus neuronas, en ella también se hace la **comparación de similitud** entre el vector de entrada y el prototipo de la neurona ganadora.
- **Capa de salida (F2):** es una capa de neuronas competitivas, o sea todas compiten para ser la ganadora, pero solo una puede serlo e inhibe a las demás.
- **Parámetro de vigilancia (p):** Dice cuan semejante debe ser la entrada con la categoría seleccionada. Este parámetro está dado por $0 < p < 1$
 - Si "p" es muy cercano a 0, muchas entradas serán categorizadas en una misma categoría (recuerdos generales)
 - Si "p" es muy cercano a 1, se crearán muchas categorías (recuerdos detallados, muy diferentes, memorización).
- **Sistema de orientación:** Sirve para orientar la red, las neuronas de ambas capas están totalmente interconectadas y hay una afluencia hacia adelante y hacia atrás.
- **Sistema de reinicio:** Sirve para inhibir la neurona ganadora cuando no cumple con la **vigilancia**, en el proceso de **comparación de similitud**.



3.2 REDES DE NEURONAS NO SUPERVISADAS

ART

- Se puede describir la acción de la red ART en términos de la actividad de cada capa para una serie de fases:
 - Fase de Inicialización.
 - Fase de Reconocimiento.
 - Fase de Comparación.
 - Fase de Búsqueda.



3.2 REDES DE NEURONAS NO SUPERVISADAS

ART

- **FASE DE INICIALIZACIÓN:** consiste en inicializar los diferentes parámetros de la red y determinar cómo serán las señales control.
 - Los pesos W y T son los mismos aunque W son una versión normalizada de T .
 - Los pesos T representan los prototipos de cada neurona.
 - Normalmente, se inicializan siguiendo el siguiente criterio:
 - $w_{ji} = 1/(1+N)$, $N = n^\circ$ entradas (pesos conexiones hacia adelante $F1 \rightarrow F2$)
 - $t_{ij} = 1$ (pesos conexiones hacia atrás - prototipos).
- **Señales de Control:** Las señales de control (orientación y reinicio) se utilizan para controlar el flujo de datos a través de la red y conmutar entre las diferentes fases:
 - **Control de Orientación:** controla el flujo para la capa de entrada y conmuta entre Entrada de Datos y Comparación.
 - **Control de Reinicio:** controla si se modifican los pesos de la ganadora o se inhibe la neurona ganadora actual cuando no satisface el criterio de vigilancia para buscar otra ganadora.

3.2 REDES DE NEURONAS NO SUPERVISADAS

ART

- **FASE DE RECONOCIMIENTO:** se efectúa una operación con el patrón de entrada y los pesos W asociados a cada neurona de la capa de salida, el resultado de esta operación debe indicar qué categoría tiene mayor prioridad para ver si el patrón entra en resonancia con ella.
 - Por ejemplo, se podría calcular la distancia euclídea entre los datos de entrada y los pesos W .
 - La clase ganadora sería aquella cuyo vector de pesos W estuviese más cerca de los datos de entrada y por tanto, sería la primera a la que se le intentaría asociar dicho patrón.
 - *Por estar mas cerca no significa que sea muy parecida, si no que es la más parecida. Luego se compara según vigilancia.*

3.2 REDES DE NEURONAS NO SUPERVISADAS

ART

- **FASE DE COMPARACIÓN:** El objetivo es obtener una medida de similitud entre el vector de entrada y el vector prototipo que surge de la capa de salida:
 - El vector de entrada y el vector prototipo producido por los pesos *feedback* de F2 (T), son comparados en la capa de entrada.
 - Se calcula el grado o % de semejanza entre la entrada y el prototipo almacenado y se comprueba si es mayor o igual que p (vigilancia).
 - Si es así, se puede concluir que dicha entrada forma parte de la categoría seleccionada y se debe proceder a **ajustar los pesos que van tanto hacia adelante (W) como hacia atrás (T)**.
 - De lo contrario se envía una **señal de reinicio**, para que inhiba esa neurona ganadora y proceda de nuevo la selección de otra ganadora, excluyendo la neurona inhibida.
- El ajuste de los pesos (W) y (T) de la ganadora, se suele realizar:
 - T: adaptándolos a la entrada, (Ej. Aplicando la regla de Hebb, como en SOM).
 - W: se obtienen normalizando los anteriores.

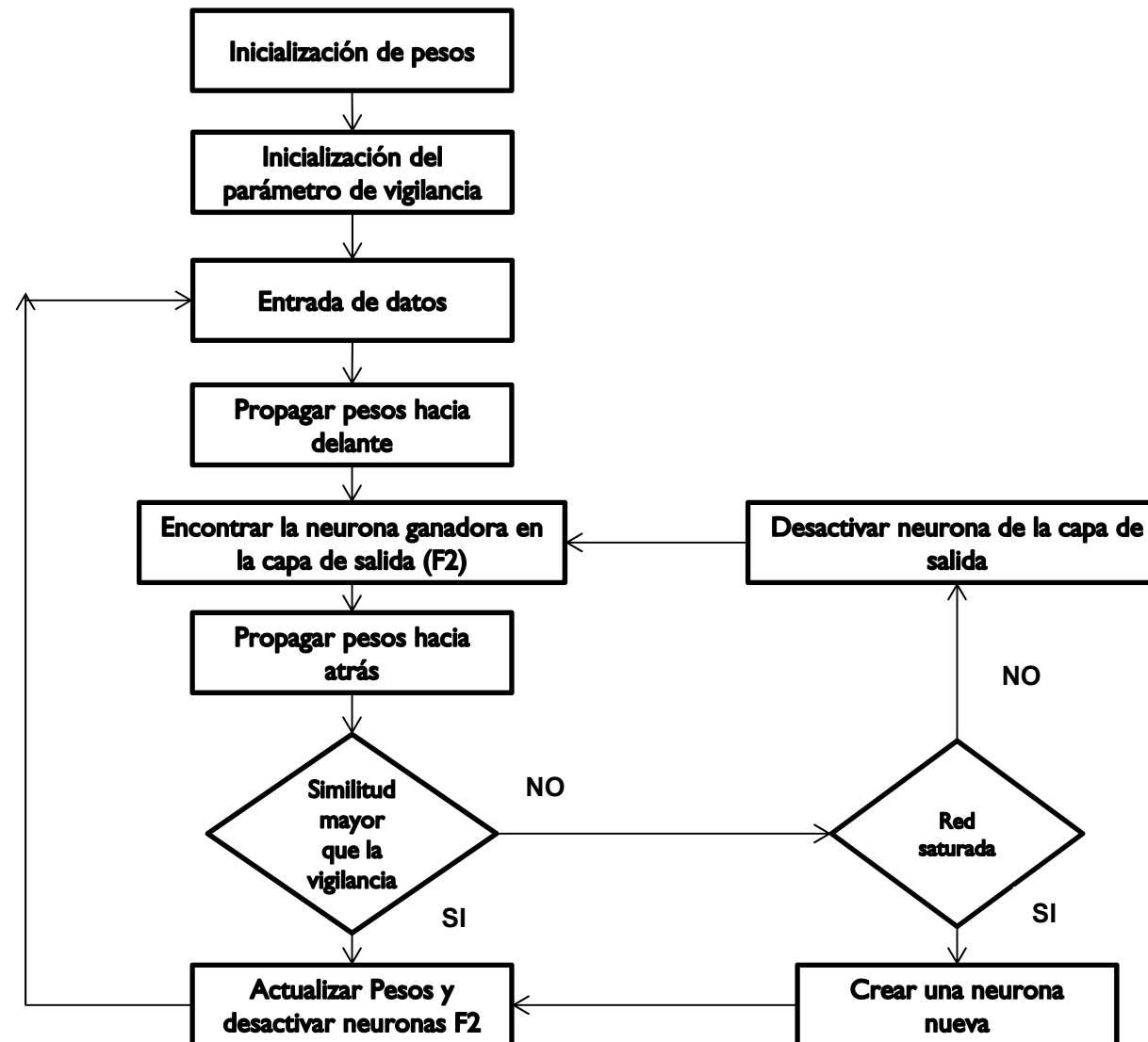
3.2 REDES DE NEURONAS NO SUPERVISADAS

ART

- **FASE DE BÚSQUEDA:** de no representar la neurona ganadora la categoría del vector de entrada, esta neurona se desactiva y se empieza la búsqueda por otras categorías que ya posea la red.
 - Se repiten entonces los pasos anteriores hasta que se encuentre una neurona ganadora que represente la categoría del vector de entrada.
 - Si se repitiera el proceso hasta que no quedara ninguna neurona se llegaría a una situación de saturación de la red que podría solucionarse ampliando el número de neuronas de la red de forma dinámica.
 - Se crearía así otra categoría nueva.
- El funcionamiento de ART se acerca bastante al funcionamiento del cerebro a la hora de categorizar/memorizar.
- Es un aprendizaje ON LINE porque no se distingue entre entrenamiento y funcionamiento.
 - Los pesos varían durante el funcionamiento de la red cuando se aplica una información de entrada a la misma.

3.2 REDES DE NEURONAS NO SUPERVISADAS

ART



3.2 REDES DE NEURONAS NO SUPERVISADAS

ART

- **LIMITACIONES:**

- Dependencia del tipo y orden de los patrones aprendidos.
- Influencia del parámetros de vigilancia (p) en el número de categorías creadas.

- **APLICACIONES:**

- Reconocimiento visual de objetos (blancos aéreos, minas); Reconocimiento de imágenes y texturas, caracteres. ART 1
- Procesamiento de señales (ECG - arritmias), reconocimiento de voz; Clasificación de patrones médicos, diagnóstico de enfermedades. ART2
- Detección de sustancias tóxicas, olores. Fuzzy ARTmap.

3.2 REDES DE NEURONAS NO SUPERVISADAS

ART

- Aplicación: reconocimiento/almacenamiento de caracteres
 - La figura muestra una sesión de entrenamiento de una red ART. A la red se le muestra letras que están representadas como pequeños cuadrados de una rejilla de 8x8.
 - En la parte izquierda se representa el conjunto de vectores de entrada y en la parte derecha se representan los patrones almacenados.

