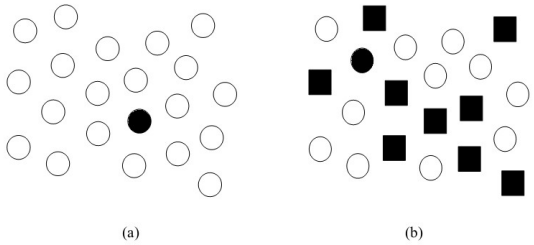
Agrupamiento (Clustering)

Aprendizaje Maquinal

FI-UNER

Alejandro Hadad
alejandro.hadad@uner.edu.ar

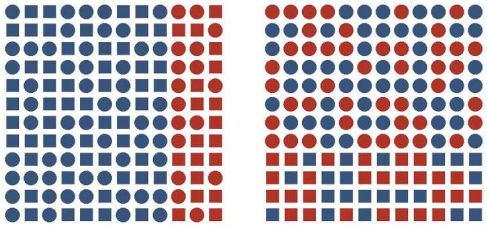
13 de Octubre de 2022



(a) (b)

Figure 1: Examples of two target detection tasks: (a) target can be detected preattentively because it possess the feature "filled"; (b) target cannot be detected preattentively because it has no visual feature that is unique from its distractors.

High-Speed Visual Estimation Using Preattentive Processing CHRISTOPHER G. HEALEY, KELLOGG S. BOOTH, and JAMES T. ENNS. The University of British Columbia. 1996

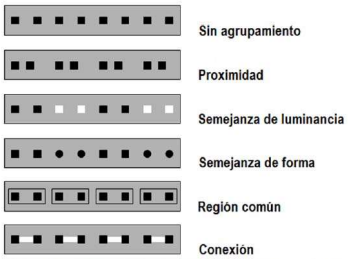


(a) (b)

Figure 2: Region segregation by form and hue: (a) hue boundary is identified preattentively, even though form varies randomly in the two regions; (b) random hue variations interfere with the identification of a region boundary based on form.

High-Speed Visual Estimation Using Preattentive Processing CHRISTOPHER G. HEALEY, KELLOGG S. BOOTH, and JAMES T. ENNS. The University of British Columbia. 1996

Principios de Agrupamiento en la Percepción Visual (Palmer & Rock, 1994)



Agrupamiento

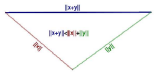
- Los resultados obtenidos dependerán de:
- El algoritmo de agrupamiento seleccionado.
- El conjunto de datos disponible.
- La medida de similitud utilizada para comparar objetos(usualmente, definida como medida de distancia).

Medidas de Similitud

- Importante tanto en agrupamiento como en clasificación
- Datos similares tendrán clases/grupos similares
- Los algoritmos empleados pueden dar resultados diferentes de acuerdo a la métrica empleada
- Diferentes formas de calcularla dependiendo del tipo de atributos
- Ej. Hamming, Euclidea, Tchebyschev, City-Block

Medidas de Similitud

- A la hora de calcular la similitud entre dos objetos/muestras
- No tienen porqué utilizarse todos los atributos disponibles en nuestro conjunto de datos
- Hay que tener cuidado con las magnitudes de cada variable → Normalización
- *Propiedades*:
- Propiedad reflexiva: d(i,j) = 0 si y solo si i=j
- Propiedad simétrica: d(i,j) = d(j,i)
- Desigualdad triangular: d(i,j) <= d(i,k) + d(k,j)

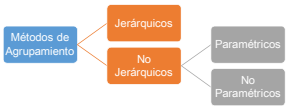


Medidas de Similitud

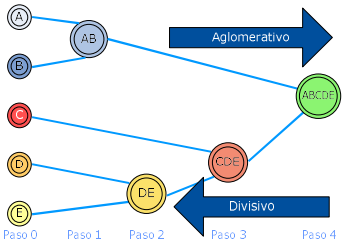
- **Distancia Euclídea**: Es la distancia clásica, como la longitud de la recta que une dos puntos en el espacio euclídeo: $d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
- **Distancia de Manhattan** o distancia por cuadras (**city-block**): Como su nombre indica, hace referencia a recorrer un camino no en diagonal (por el camino más corto), sino zigzagueando: $d(x,y) = \sum_{i=1}^{n}|x_i - y_i|$
- **Distancia de Tchebychev**: Simplemente calcula la discrepancia más grande en alguna de las dimensiones: $d(x,y) = \max_{i=1..n}|x_i - y_i|$
- **Distancia del coseno:** Si se considera que cada ejemplo es un vector, la distancia sería el coseno del ángulo que forman: $d(x,y) = \arccos\left(\frac{x^T y}{\|x\| \cdot \|y\|}\right)$
- **Distancia de Mahalanobis:** Todas las distancias anteriores asumen, en cierto modo, que los atributos son independientes (es decir, consideran cada atributo una dimensión ortogonal a las demás). Una distancia más robusta es ésta, que utiliza la matriz de covarianzas S: $d(x,y) = \sqrt{(x-y)^T S^{-1}(x-y)}$

Algoritmos de Agrupamiento

- Se buscan agrupamientos naturales en los datos tal que los individuos de cada grupo muestren semejanzas.
- Aprendizaje no supervisado.

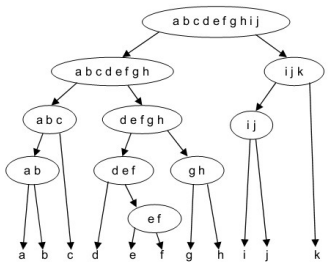


Clustering Jerárquico - Dendogramas



Clustering Jerárquico - Dendogramas

- ***En la figura existe una jerarquía de diez niveles***:
- Nivel 1: Conjunto a,b,c,d,e,f,g,h,i,j,k
- Nivel 2: Conjunto a,b,c,d,e,f,g,h
- Nivel 3: Conjunto i,j,k
- Nivel 4: Conjunto i,j
- Nivel 5: Conjunto a,b,c
- Nivel 6: Conjunto d,e,f,g,h
- Nivel 7: Conjunto d,e,f
- Nivel 8: Conjunto a,b
- Nivel 9: Conjunto g,h
- Nivel 10: Conjunto e,f



Clustering Jerárquico

- Dependiendo de la manera de construir el árbol los métodos se dividen en:
- **Aglomerativos**. El árbol se va construyendo empezando por las hojas, hasta llegar a la raíz. En un primer momento cada ejemplo es a su vez un grupo, se van aglomerando los grupos para formar conjuntos cada vez más numerosos, hasta llegar a la raíz, que contiene a todos los ejemplos.
- **Divisivos**. Se parte de la raíz, que es un solo grupo conteniendo a todos los ejemplos, y se van haciendo divisiones paulatinas hasta llegar a las hojas que representa a la situación en que cada ejemplo es un grupo.

Clustering Jerárquico

- Los métodos aglomerativos parten de dos principios fundamentales.
- La *forma de seleccionar los grupos a mezclar*, y la *manera de mezclarlos*.
- El método más común es elegir aquellos grupos cuya *distancia de enlace* (link distance) sea *menor*.
- Una manera de hacerlo es obligar a que *cada grupo tenga un representante* (que puede crearse como un centroide) que será utilizado como elemento de referencia para el cálculo de las distancias.
- La mezcla de grupos consiste en hacer que *todos los ejemplos de los grupos que se van a mezclar pasen a ser miembros del nuevo grupo*.
- En cuanto al *representante del nuevo grupo creado suele ser el centro de masas* de los puntos pertenecientes a la clase.

Clustering Jerárquico

- El método final quedaría como sigue:

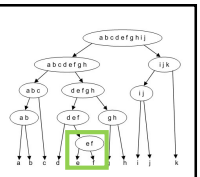
1. Hacer que cada punto sea el representante de un grupo que sólo contiene dicho punto.
2. Calcular las distancias entre todos los grupos existentes dos a dos.
3. Elegir los dos grupos cuya distancia sea menor.
4. Mezclar los grupos elegidos en el paso anterior. Si el representante de uno de los grupos es el vector $\vec{C}_a = \{c_{a_1}, c_{a_2}, ..., c_{a_n}\}$, y el del otro grupo $\vec{C}_b = \{c_{b_1}, c_{b_2}, ..., c_{b_n}\}$ si además el grupo **a** tiene **j** ejemplos y el grupo **b** tiene **k** ejemplos; el nuevo representante se calculará mediante la expresión:
4. Si hay más de un grupo ir a 2.

$$\vec{C} = \left\{ \frac{j \cdot c_{a_1} + k \cdot c_{b_1}}{j+k}, \frac{j \cdot c_{a_2} + k \cdot c_{b_2}}{j+k}, ..., \frac{j \cdot c_{a_n} + k \cdot c_{b_n}}{j+k} \right\}$$

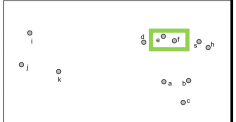
Clustering Jerárquico

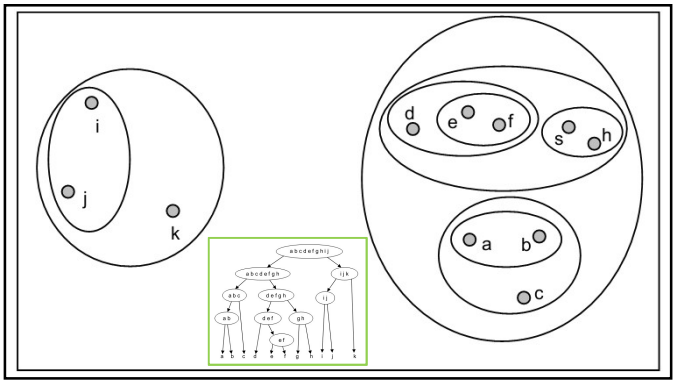
- Dependiendo de cómo se calcule la distancia de enlace entre grupos se pueden distinguir tres métodos:
- *Enlace simple* (***single linkage***): para el cálculo de la distancia no se utilizan los representantes, sino que se calcula la distancia entre todos los puntos de dos grupos y se toma como distancia entre grupos la ***menor***.
- *Enlace completo* (***complete linkage***): igual que el anterior, pero se toma como distancia entre grupos la ***mayor*** de todas.
- *Enlace en la media* (***average linkage***): se toma como distancia la existente entre los ***representantes*** (***centroides***) de los grupos.

	a	b	c	d	e	f	g	h	i	j	k
a	-	2,4	4	9	7,8	7	8,9	8,7	27,6	28,3	21,4
b	-	-	4,3	11,4	9,1	7,2	7,3	7,4	30,8	32,6	26,8
c	-	-	-	13,6	12,2	11	12,6	10,5	31	31,9	25,2
d	-	-	-	-	4,8	6,4	10,9	12	21	23,7	17,6
e	-	-	-	-	-	1,7	6	7,4	26,3	28,7	22,5
f	-	-	-	-	-	-	4,5	5,6	27,3	30	23,6
g	-	-	-	-	-	-	-	1,8	31,9	34,3	28,8
h	-	-	-	-	-	-	-	-	33,4	35,4	28,8
i	-	-	-	-	-	-	-	-	-	4	6,8
j	-	-	-	-	-	-	-	-	-	-	6,9
k	-	-	-	-	-	-	-	-	-	-	-



	a	b	c	d	ef	g	h	i	j	k
a	-	2,4	4	9	7,4	8,9	8,7	27,6	28,3	21,4
b	-	-	4,3	11,4	8,5	7,3	7,4	30,8	32,6	26,8
c	-	-	-	13,6	11,6	12,6	10,5	31	31,9	25,2
d	-	-	-	-	5,6	10,9	12	21	23,7	17,6
ef	-	-	-	-	-	5,2	6,4	26,8	29,1	22,9
g	-	-	-	-	-	-	1,8	31,9	34,3	28,8
h	-	-	-	-	-	-	-	33,4	35,4	28,8
i	-	-	-	-	-	-	-	-	4	6,8
j	-	-	-	-	-	-	-	-	-	6,9
k	-	-	-	-	-	-	-	-	-	-





Clustering Jerárquico

Fortalezas

- Simple de implementar, usar y entender.
- No asume un número particular de clusters.
- Directamente asociado a ejemplos prácticos (ej. Árboles filogenéticos).

Debilidades

- El costo computacional es alto:

$O(n^2)$ en espacio (se debe calcular la matriz de dist.)

$O(n^3)$ en tiempo (n pasos y en cada uno se actualiza la matriz de distancia)

Clustering no Jerárquico -Algoritmos

Paramétricos

- **EM** (expectación-maximización) (Dempster et al. 1977)

No Paramétricos

- **k-means**
- **Mapas Autoorganizados de Kohonen**
- **Fuzzy C-Means**
- **Cobweb** (Fisher 1987)
- **AUTOCLASS** (Cheeseman & Stutz 1996)

K-means

El algoritmo K medias (del inglés Kmeans) se trata de un ***método de agrupamiento por vecindad*** en el que se parte de un número determinado de ***prototipos*** y de un conjunto de ejemplos a agrupar, sin etiquetar.

Es el método más popular de los métodos de agrupamiento denominados "por partición", en contraposición de los métodos jerárquicos. La idea del K medias es situar a los prototipos o centros en el espacio, de forma que los datos pertenecientes al mismo prototipo tengan características similares [Moody & Darken 1989, MacQueen 1967].

Todo ejemplo nuevo, una vez que los prototipos han sido correctamente situados, es comparado con éstos y asociado a aquél que sea el más próximo, en los términos de una distancia previamente elegida. Normalmente, se usa la distancia euclídea.

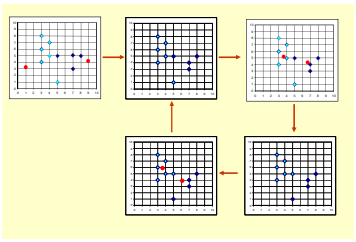
K-means

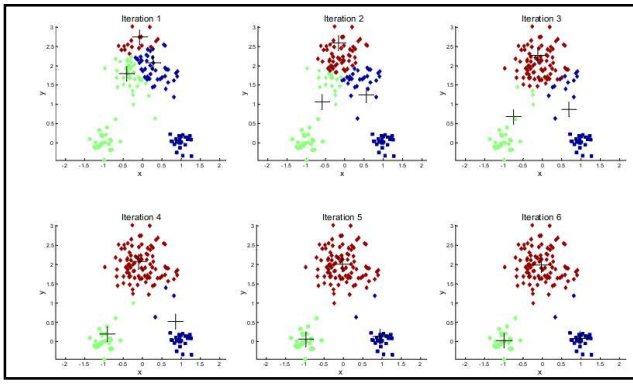
El método tiene una fase de entrenamiento, que puede ser lenta, dependiendo del número de puntos a clasificar y de la dimensión del problema. Pero una vez terminado el entrenamiento, la clasificación de nuevos datos es muy rápida, gracias a que la comparación de distancias se realiza sólo con los prototipos.

El procedimiento es el siguiente:

- Se calcula, para cada ejemplo **xk**, el prototipo más próximo **Ag** y se incluye en la lista de ejemplos de dicho prototipo. $A_g = \arg\min_{A_i} \{d(x_k, A_i)\}$
- Después de haber introducido todos los ejemplos, cada prototipo Ak tendrá un conjunto de ejemplos a los que representa: $l(A_k) = \{x_{k_1}, x_{k_2}, ..., x_{k_m}\}$
- Se desplaza el prototipo hacia el centro de masas de su conjunto de ejemplos → $A_k = \frac{\sum_{i=1}^{m} x_{k_i}}{m}$
- Se repite el procedimiento hasta que ya no se desplazan los prototipos.

K-Means





K-Means

Fortalezas

• Relativamente eficiente: O(tkn)

donde n: Nº de objetos, k: Nº clusters, t: Nº iteraciones

Debilidades

•Aplicable solamente cuando la media esté definida.

•Se debe especificar el k de antemano

•Presenta problemas con clusters de distintos tamaño y densidades y morfologías no convexas

•Poco robusto a datos ruidosos y outliers

•Frecuentemente obtiene óptimos locales

Observaciones

• Datos categóricos: se reemplaza media por moda y se usan otras medidas de similitud

Calidad de los Clusters

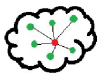
- Una vez seleccionado el número adecuado de clusters y aplicado el algoritmo de clustering pertinente se tiene que evaluar la calidad de los de los mismos, de lo contrario, podrían derivarse conclusiones de agrupación que no se corresponden con la realidad.
- Pueden diferenciarse tres tipos de estadísticos empleados con este fin:
- **Validación interna** de los clusters: Emplean únicamente información interna del proceso de clustering para evaluar la bondad de las agrupaciones generadas. Se trata de un proceso totalmente **no supervisado** ya que no se incluye ningún tipo de información que no estuviese ya incluida en el clustering.
- **Validación externa** de los clusters (ground truth): Combinan los resultados del clustering (no supervisado) con información externa (**supervisado**), como puede ser un set de validación en el que se conoce el verdadero grupo al que pertenece cada observación. Permiten evaluar hasta qué punto el clustering es capaz de agrupar correctamente las observaciones. Se emplea principalmente para seleccionar el algoritmo de clustering más adecuado, aunque su uso está limitado a escenarios en los que se dispone de un set de datos de validación.

Silhouette width

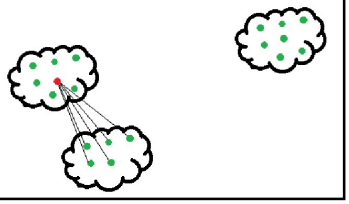
- Cuantifica cómo de buena es la asignación que se ha hecho de una observación comparando su similitud con el resto de observaciones del mismo cluster frente a las de los otros clusters. Para cada observación **i**, el *silhouette coeficient* (**si**) se obtiene del siguiente modo:
- Calcular la **media de las distancias** (llámese **ai**) entre la ***observación* i** y el *resto de observaciones que pertenecen al mismo cluster*. Cuanto *menor* sea *ai mayor* la *similitud* que tiene con el resto de observaciones de su cluster → **COHESION**
- Calcular la distancia promedio entre la observación **i** y el resto de clusters. Entendiendo por distancia promedio entre **i** y un determinado cluster **C** como *la media de las distancias entre i y las observaciones del cluster C*.
- Identificar como **bi** a la *menor de las distancias promedio* entre i y el resto de clusters, es decir, la **distancia al cluster más próximo** (neighbouring cluster) → **SEPARACIÓN**
- Calcular el valor de silhouette como: $$s(x) = \frac{b(x) - a(x)}{\max\{a(x), b(x)\}}$$

Silhouette width

• COHESION →



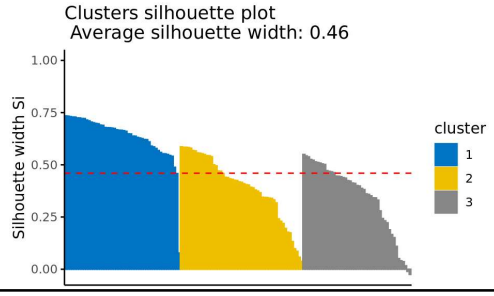
• SEPARACIÓN →



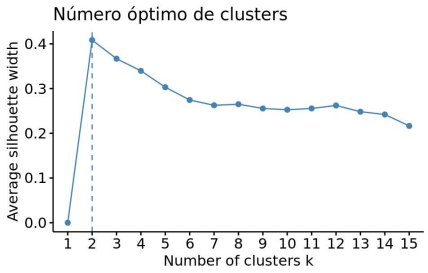
Silhouette width

- Su valor puede estar entre -1 y 1, siendo **valores altos** un indicativo de que la observación se ha asignado al ***cluster* correcto**. Cuando su valor es próximo a cero significa que la observación se encuentra en un punto intermedio entre dos *clusters*. Valores negativos apuntan a una posible asignación incorrecta de la observación. Se trata por lo tanto de un método que permite evaluar el resultado del *clustering* a múltiples niveles:
- La **calidad de asignación** de cada observación por separado. Permitiendo identificar potenciales asignaciones erróneas (valores negativos de *silhouette*).
- La **calidad de cada *cluster*** a partir del promedio de los índices *silhouette* de todas las observaciones que lo forman. Si por ejemplo se han introducido demasiados *clusters*, es muy probable que algunos de ellos tengan un valor promedio mucho menor que el resto.
- La **calidad de la estructura de *clusters*** en su conjunto a partir del promedio de todos los índices *silhouette*.

Silhouette width



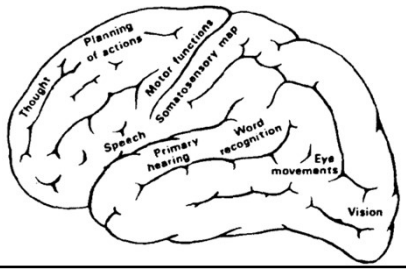
Silhouette width



Índice Dunn

- El índice *Dunn* es otra medida de validación interna que se obtiene de la siguiente forma:
- Para cada *cluster* calcular la distancia entre cada una de las observaciones que lo forman y las observaciones de los otros *clusters*.
- Seleccionar como "representante" de la distancia entre *clusters* a la menor de todas las distancias calculadas en el paso anterior (**separación mínima *inter-clusters***).
- Para cada *cluster* calcular la distancia entre las observaciones que lo forman (*intra-cluster distance*).
- Seleccionar como "representante" de la distancia *intra-cluster* a la mayor de todas las distancias calculadas en el paso anterior (**separación máxima *intra-cluster***).
- Calcular el índice *Dunn* como:
  - **D = separacion minima interclusters / separacion maxima intracluster**
- Si la estructura contiene clusters compactos y bien separados, el numerador es grande y el denominador pequeño, dando lugar a valores altos de D.
- El objetivo por lo tanto es maximizar el índice Dunn.
- ***Inconveniente***: Si todos los clusters tienen un comportamiento ideal excepto uno, cuya calidad es baja, dado que el denominador emplea el máximo en lugar de la media, el índice estará totalmente influenciado por este cluster enmascarando al resto

Mapas de auto organización



Introducción

Fueron creadas en 1982 por el Prof. Tehuvo Kohonen, por lo que se conocen también como mapas de Kohonen.

Los mapas de auto organización (Self Organizing Maps, SOM) son redes neuronales antero-alimentadas (feedforward) cuyo entrenamiento es del tipo no supervisado utilizando el paradigma competitivo.

Estas redes descubren rasgos comunes, regularidades, correlaciones o categorías en los datos de entrada, e incorporarlos a su estructura interna de conexiones. Se dice, por tanto, que las neuronas deben auto-organizarse en función de los estímulos (datos) procedentes del exterior.

Introducción

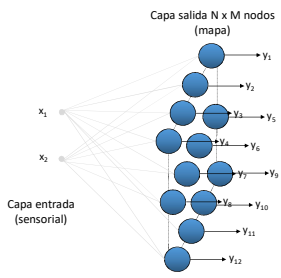
Mapas sensoriales

En las cortezas cerebrales sensoriales se ha observado que la influencia que una neurona ejerce sobre las demás es función de la distancia entre ellas, siendo muy pequeña cuando están muy alejadas [Regla de Hebb].

Considerando esto, los SOM pretenden mimetizar de forma simplificada la capacidad del cerebro de formar mapas topológicos a partir de las señales recibidas del exterior

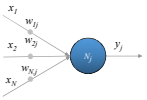
Definición

Estructura



Definición | Estructura

Nodo de un SOM:



$$y_j = \|x - w\| = \sum_{i=1}^{N} (x_i - w_{ij})^2$$

Aprendizaje competitivo

Este paradigma se basa en la existencia de una cierta **competitividad** entre los nodos de la red neuronal de una cierta capa por la oportunidad de entrenarse (aprender).

Puntualmente, el nodo que produce la salida mayor (o menor, depende del tipo de salida) se le considera **ganador**, y tiene la capacidad de inhibir a los otros nodos (no presentan activación: salida nula).

Los nodos se relacionan selectivamente con las distintas entradas presentadas a la red, a fin de lograr un ordenamiento entre las distintas neuronas. De manera de definen zonas de nodos activos asociadas a las diferentes características relevantes de las entradas.

Aprendizaje competitivo

SOM de 6 x 6 nodos.

Aprendizaje competitivo | Realimentación lateral

La realimentación lateral, existente en las neuronas biológicas, se define como una realimentación entre neuronas, que depende de la distancia lateral entre las mismas.

Influye en la modificación de los pesos de una neurona, ya que esto involucrará a neuronas *"vecinas"* dentro de la estructura de la red.

Aprendizaje competitivo | Realimentación lateral

Este tipo de realimentación tiene las siguientes características:

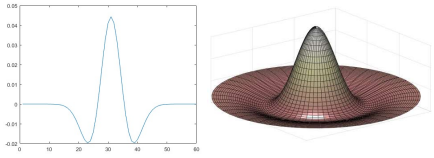
✓ una primera zona de excitación centrada en la neurona en consideración.

✓ una segunda zona de inhibición, contigua a la anterior.

✓ una tercera zona de muy baja excitación (generalmente no considerada).

Aprendizaje competitivo | Realimentación lateral

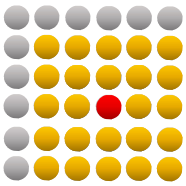
Función sombrero mexicano



Emular el efecto de la *realimentación lateral* es muy complicado, por lo que aparece el concepto de *región de vecindad*.

Aprendizaje competitivo

Región de vecindad



Algoritmo "Self- Organizing Maps"- SOM

El principal objetivo del algoritmo SOM es transformar señales patrones de entrada de una dimensión arbitraria, en una representación discreta de dimensión uno o dos, realizando esta transformación en forma adaptativa y ordenada topológicamente.

Los patrones son presentados a la red de a uno por vez, cada presentación provoca que un grupo localizado de neuronas en la red sean activadas.

Los componentes principales que intervienen en el algoritmo son:

- ✓ un reticulado de neuronas (máxima dimensión: 2).
- ✓ un mecanismo que compare determinadas funciones discriminantes de las entradas y determine las neuronas que mejor verifiquen estas funciones, definiendo de tal forma las neuronas ganadoras.
- ✓ un proceso adaptativo que permite a las neuronas activadas, mejorar la función que las relaciona a las entradas.

Desarrollo del algoritmo

Sea x un vector patrón de entrada:

$$x = [x_1, x_2, \dots \dots \dots . x_p]$$

El vector de pesos de la neurona j viene dado por:

$$w_j = [w_{j1}, w_{j2}, \dots \dots \dots . w_{jp}] \qquad j = 1,2 \dots \dots . . N$$

Teóricamente, para determinar el mejor *"matching"* de las neuronas con el vector de entrada, se computa el producto:

$$x \cdot w_j^T \ , \qquad j = 1,2 \dots \dots . . N$$

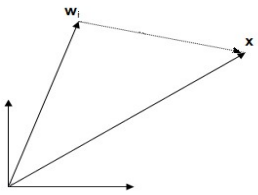
Y luego se selecciona el mayor.

De esta manera se determina la neurona con mayor excitación externa y por lo tanto la localización de la neurona ganadora.

Sin embargo en la práctica es conveniente utilizar un criterio de *"matching"* que permita normalizar los vectores de pesos.

El criterio utilizado en este caso es la distancia Euclídea entre los vectores, buscando determinar la distancia mínima:

$$D_i = |x - w_i|$$

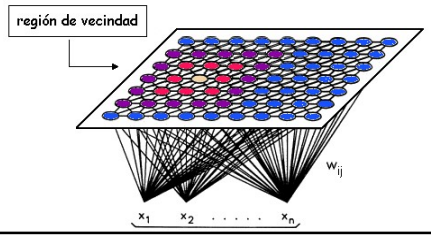


Determinada la neurona con la mínima distancia, ésta representa al patrón de entrada presentado, de esta manera datos de entrada que pueden ser de una dimensión mayor, son representados por un vector de dos dimensiones (máximo).

Dependiendo de la aplicación, la respuesta de la red puede ser:

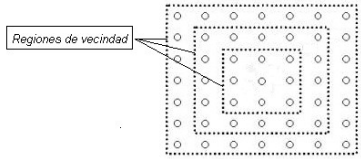
- ✓ la posición de la neurona ganadora en el reticulado.
- ✓ el vector de pesos de la neurona ganadora.

Para determinar la región de neuronas vecinas a la ganadora, que conjuntamente representan al patrón de entrada, se utiliza la *Región de Vecindad* $\Lambda_{i(x)}(n)$, que determina la vecindad de la neurona ganadora.



Se adopta un valor grande de la función $\Lambda_{i(x)}(n)$ al comenzar el proceso iterativo, para luego hacerlo disminuir con el paso de las iteraciones. Siempre centrada en las diferentes neuronas ganadoras.

Evolución de la región de vecindad:



Regla de Hebb [1949]

- La regla refería a cierto comportamiento biológico que Hebb observó en las células cerebrales
- Regla de Hebb: *Cuando un axón de una célula A está lo suficientemente cerca para excitar a una célula B, y toma parte repetidamente en el proceso de disparo de dicha célula, se produce algún tipo de cambio metabólico en una de las células (o en las dos),* ***que hace que la eficacia con la que A dispara a B se vea incrementada****.*

Durante el entrenamiento la neurona con la mínima distancia, en conjunto con las neuronas vecinas, ajustan sus pesos con el objetivo de acercarse aún más a los valores de las entradas.

Para implementar este proceso de ajuste se utiliza el siguiente proceso de aprendizaje de *Hebb* modificado:

$$\frac{dw_j}{dt} = \begin{cases} \eta(x - w_j) & \text{para neuronas dentro de } \Lambda_{i(x)}(n) \\ 0 & \text{para neuronas fuera de } \Lambda_{i(x)}(n) \end{cases}$$

Finalmente el proceso de adaptación de los pesos es:

$$w_j(n+1) = \begin{cases} w_j(n) + \eta(n)\left(x - w_j(n)\right) & j \varepsilon \Lambda_{i(x)}(n) \\ w_j(n) & \text{resto} \end{cases}$$

Para el parámetro de aprendizaje $\eta(n)$ se recomienda comenzar (primeras 1000 iteraciones) con un valor mayor (siempre menor a la unidad), para luego reducirlo . La forma de variación de $\eta(n)$ no es crítica.

La primera fase del proceso de adaptación se denomina **fase de ordenamiento** y es donde se produce el verdadero ordenamiento de los vectores de pesos $w_j(n)$.

La segunda fase del proceso denominada **fase de convergencia**, más larga que la anterior, produce un ajuste fino de los pesos, donde el valor de $\eta(n)$ puede tomar valores muy bajos (0.01).

En cuanto a la Región de Vecindad $\Lambda_{i(x)}(n)$, puede tomar diferentes formas: cuadrada, romboidal o gaussiana.

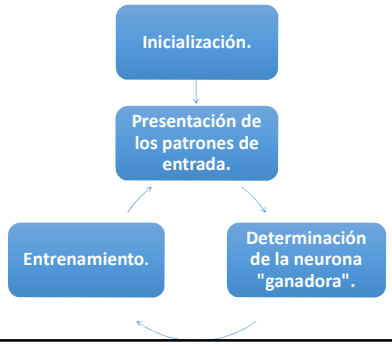
Su valor comienza generando una zona que puede contener todas las neuronas de la red, para luego disminuir su radio en función del tiempo. Esta disminución se puede implementar en forma lineal durante las primeras iteraciones.

Durante la fase de convergencia $\Lambda_{i(x)}(n)$ debe contener solamente las neuronas vecinas a la neurona ganadora, para concluir el entrenamiento incluyendo solamente a esta última.

Resumen del algoritmo.

Lo esencial del algoritmo es permitir implementar en forma relativamente sencilla, consideraciones más complejas como la regla modificada de *Hebb* y la realimentación lateral.

Consta de las siguientes etapas:



Algoritmo de entrenamiento

a) Inicializar los pesos aleatoriamente **W**entre (-0.5 , 05).

b) Escoger un patrón del conjunto de entrenamiento **x**, calcular la salida de cada nodo y determinar el **nodo ganador** N_j.

c) Adaptar los pesos de los nodos que se encuentran dentro de la vecindad alrededor del nodo ganador N_j según la siguiente ecuación:

$$w_{ij}(n+1) = w_{ij}(n) + \eta.\|x - w_{ij}(n)\|$$

donde $\eta = 1 - \frac{n-1}{N}$ y n es el índice de iteración y N la cantidad total de iteraciones.
Para los nodos afuera de la vecindad la ecuación es la siguiente:

$$w_{ij}(n+1) = w_{ij}(n)$$

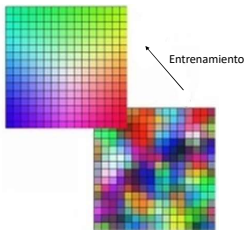
d) Volver al paso b) y repetir el proceso hasta completar las totalidad de las iteraciones.

Aprendizaje competitivo | Resumen

El objetivo del algoritmo de aprendizaje de SOM es almacenar una serie de patrones de entrada $\mathbf{x} \in \mathbf{X}$, a través de encontrar un conjunto de prototipos $\mathbf{w_j}$, con j = 1, 2…M y M es la cantidad de nodos de la red. A este conjunto se lo denomina Φ.

Este conjunto Φ representa al mejor mapa de características posible presenta alguna estructura topológica..

Algoritmo de entrenamiento | Auto organización



Algoritmo de entrenamiento | Etiquetado

Una vez entrenada la red SOM se procede al etiquetado de la misma, para lo cual se le presentan a la red todos los casos y se identifican las neuronas que representan cada caso.

Los limites entre los diferentes grupos que la red entrenada genera, se determinan comparando las distancias entre los vectores de pesos de las neuronas etiquetadas. Si la distancia es reducida pertenecen al mismo grupo, en cambio si la distancia es grande pertenecen a grupos diferentes.

Algoritmo de entrenamiento | Recomendaciones

La región de vecindad puede tomar diferentes formas: cuadrada, romboidal o gaussiana.

Su valor comienza generando una zona que puede contener todas las neuronas de la red, para luego disminuir su radio en función de las iteraciones. Esta disminución se puede implementar en forma lineal durante las primeras 1000 iteraciones.

Durante la fase de convergencia, la región de vecindad debe contener solamente las neuronas vecinas a la neurona ganadora, para concluir el entrenamiento incluyendo solamente a esta última.

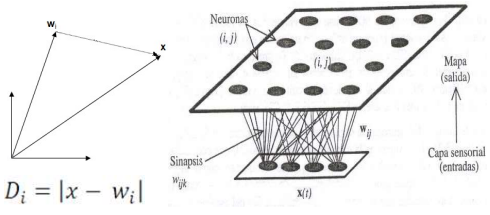
INTERPRETACION DE LOS RESULTADOS

Una vez entrenada la red SOM se procede al etiquetado de la misma, para lo cual se le presentan a la red todos los casos y se identifican las neuronas que representan cada caso.

Los limites entre los diferentes grupos que la red entrenada genera, se determinan comparando las distancias entre los vectores de pesos de las neuronas etiquetadas. Si la distancia es reducida pertenecen al mismo grupo, en cambio si la distancia es grande pertenecen a grupos diferentes.

Medidas de calidad de la Red de Kohonen

- Naturalmente no surge una medida como el error cuadrático en las redes supervisadas



Medidas de calidad de la Red de Kohonen

- En este caso se define el error de cuantización y la error topológico
- Para cada patrón de entrenamiento **x** se puede calcular:

$$d(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{m}_c\| + \sum_{k=0}^{K_{c'j}-1} \|\boldsymbol{m}_{l_i(k)} - \boldsymbol{m}_{l_i(k+1)}\|.$$

- Dado el conjunto de entrenamiento $\{\boldsymbol{x_n}\}_{n=1}^{N}$ el error global de la red está dado por

$$E_C = \sum_{n=1}^{N} d(\boldsymbol{x}_n)$$

Bibliografía

- **Self-organizing maps**. Autor: Teuvo Kohonen. Editorial:Springer Verlag (1997).ISBN: 3-540-62017-6
- **Capítulo 6. Aprendizaje No Supervisado**. Redes de Neuronas Artificiales. Un Enfoque Práctico. Autores: Viñuela Leon. Editorial: Pearson Education (2004)
- **Self-organization and missing values in SOM and GTM**. Authors: T. Vatanen, M. Osmala, T. Raiko, K. Lagus, M. Sysi-Aho, M. Orešič, T. Honkela, H. Lähdesmäki. Neurocomputing Volume 147, 5 January 2015, Pages 60-70. http://users.ics.aalto.fi/praiko/papers/vatanen2014.pdf

FIN