Mapa autoorganizativo de Kohonen

Otros cursos y tutoriales: comercio electrónico, WAP, Webmaster
Página principal del grupo GeNeura

J. J. Merelo

1 Introducción

Los mapas autoorganizativos de Kohonen son un algoritmo, a veces agrupado dentro de las redes neuronales, que a partir de un proceso de entrenamiento agrupa los datos; este agrupamiento hace que la proyección de estos datos sobre el mapa distribuya sus características de una forma gradual. El Mapa de Kohonen, SOM (self-organizing map, mapa autoorganizativo) o SOFM (self-organizing feature map, mapa autoorganizado de características) se usa para diferentes aplicaciones:

- Clustering: se pueden agrupar datos del conjunto de entrada, atendiendo a diferentes criterios.
- Visualización: este agrupamiento, como se realiza de una forma ordenada, permite visualizarlo y descubrir características nuevas, o relaciones que no se habían previsto de antemano. También permite visualizar la evolución temporal de un conjunto de datos.
- Clasificación: una vez calibrado el mapa, o asignada algún tipo de etiqueta a cada clusters, se puede usar para clasificar sobre datos desconocidos.
- Interpolación de una función: asignando valores numéricos a cada uno de los nodos de la red de Kohonen, se pueden asignar esos valores numéricos a los vectores de entrada.
- Cuantización vectorial, es decir, aplicación de una entrada continua a una salida que está discretizada, es decir, obtener a partir de un vector cualquiera el vector más cercano de un conjunto previamente establecido.

Los SOMs son algoritmos no-supervisados; eso quiere decir que no usan la etiqueta de los datos para el entrenamiento; sin embargo, en la mayor parte de los casos se usa algún tipo de etiqueta para visualizar los datos correctamente. Se denominan también algoritmos competitivos porque en el entrenamiento se entrena una sola "neurona" de cada vez, lo cual significa que la representación de cada zona del espacio de entrada está concentrada por neuronas, no distribuida (como suele suceder en otras redes neuronales tales como el perceptrón multicapa).

Hasta el momento, ha habido cientos de aplicaciones de los mapas autoorganizativos de Kohonen, muchas de ellas en Biocomputación. En este tutorial se tratará de dar un enfoque práctico al uso de los mapas de Kohonen en biocomputación, aplicándolo a problemas comunes en el área y usando herramientas habituales en la misma

2 Aprendizaje no supervisado

Los algoritmos de clasificación no supervisados son aquellos que no requieren un etiquetado de cada uno de los vectores de entrada; se suelen llamar también algoritmos auto-asociativos, porque asocian entradas a ellas mismas. Una buena explicación de estos algoritmos se halla en la FAQ de redes neuronales.

El tipo más común de algoritmos de aprendizaje o clasificación no supervisada son los algoritmos de análisis de grupos o clustering; estos algoritmos tratan de dividir las muestras del conjunto de entrada en una serie de grupos con características comunes. Un algoritmo debe descubrir cuáles son estos clusters, pero también cuáles son las características que define ese cluster y cuántos clusters hay; pero éste último es un problema que no tiene solución fácil.

Dentro de las redes neuronales, uno de los métodos no supervisados más comunes es precisamente el SOM, pero hay otro método denominado aprendizaje hebbiano que consiste en aumentar el valor de los pesos que unen a dos neuronas si se activan a la vez, y disminuir el valor si se activan de forma diferencial. Una red hebbiana se puede disponer en una sola capa o varias: las entradas se propagan a la capa interna, y a la salida, y tras la propagación, se cambian los pesos de la forma indicada. El aprendizaje hebbiano equivale a un análisis de componentes principales de las entradas.

Una red neuronal supervisada tal como el perceptrón multicapa se puede convertir en no supervisada usando las entradas como salidas; de esta forma, la capa interna extraerá los

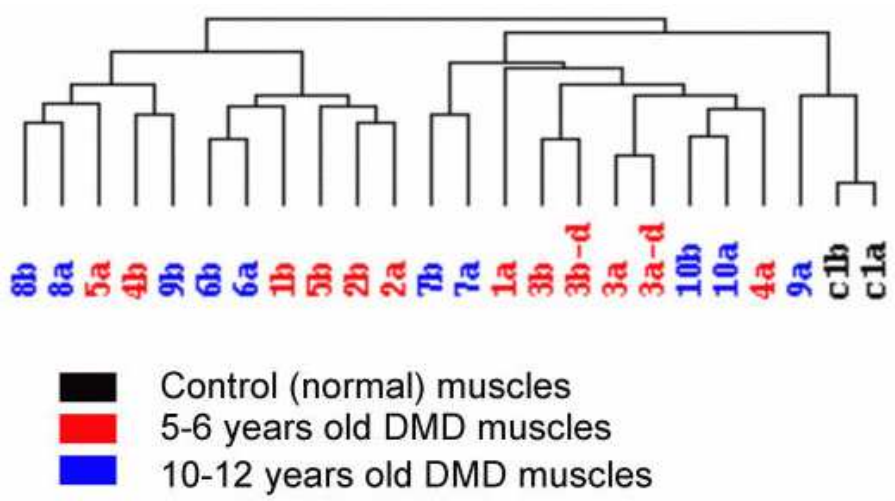
componentes principales de las entradas, y se podrá usar, por ejemplo, como memoria asociativa; o bien, analizando las activaciones de la capa interna, se pueden asignar diferentes grupos a las entradas.

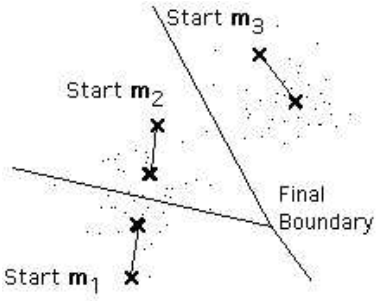
Los métodos no supervisados se suelen usar en lo denominado análisis de datos exploratorio, es decir, en una fase del análisis de los datos, cuando no se sabe de antemano cuáles son los grupos naturales que se forman, y se quiere visualizar la abundancia y la relación que hay entre los grupos "naturales"; se puede decir que una de sus principales aplicaciones es la visualización de datos multidimensionales, porque un algoritmo no supervisado actúa como una proyección de un espacio multidimensional a otro de dimensiones visualizables. También se pueden usar como fase inicial de algoritmos de aprendizaje supervisados: un algoritmo como el k-medias o el mismo SOM se pueden usar para inicializar algoritmos de aprendizaje supervisado tales como el LVQ (Learning Vector Quantization).

3 Algoritmos de clustering

Los algoritmos de clustering agrupan, de forma no supervisada, las muestras de entrada en una serie de grupos, extrayendo, a la vez, unos representantes de la muestra de entrada; a este grupo se le llama habitualmente diccionario.

Hay dos grupos de algoritmos de clustering: los jerárquicos , que van creando clusters de pequeño tamaño, incluso incialmente con un solo componente, y los van fusionando hasta obtener clusters de tamaño superior; diferentes versiones de algoritmos se diferencian por las reglas que usan para unir clusters o separarlos; el resultado final es un árbol de clusters denominado dendrograma, que muestra como los clusters se relacionan unos con otros. El clustering jerárquico se ha usado, por ejemplo, para clasificación de documentos, o incluso para clasificación en biocomputación.

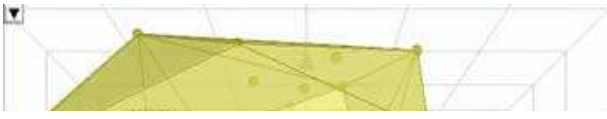




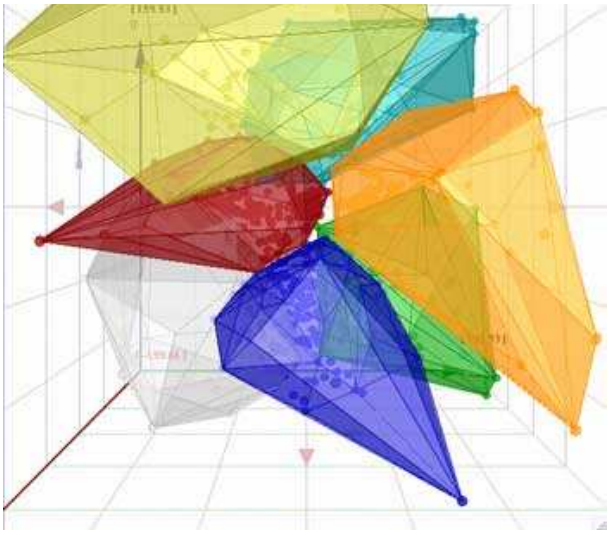
Por el contrario, el clustering no jerárquico calcula los clusters directamente; los algoritmos más populares, tales como el k-medias y el ISODATA son de este tipo. K-medias, también llamado a veces c-medias, funciona de la forma siguiente: se escoge inicialmente un número de clusters (se puede haber hallado ese número usando alguna técnica de análisis exploratorio, o, la mayor parte de las veces, a ojo de buen cubero^{TM}) Se escogen aleatoriamente el mismo número de vectores de referencia, usando alguna provisión adicional, por ejemplo, una distancia mínima entre ellos; a continuación, se asigna cada elemento de la muestra de entrada al vector de referencia más cercano. Una vez asignados todos los vectores de la muestra, se actualizan los vectores de referencia como la media de todos los vectores en un clusters. El algoritmo se repite hasta que en dos iteraciones no varíen los vectores de referencia, o, lo que es lo mismo, la distancia media de los vectores de la muestra a los vectores de referencia más cercanos o ganadores.

El algoritmo ISODATA es similar, pero el número de clusters no está fijo de antemano. Un cluster se divide si su desviación estándar excede un valor determinado, o el número de elementos de un cluster es dos veces un valor predeterminado; dos clusters se fusionan si su distancia es inferior a un umbral, o bien si el número de elementos que contienen es inferior a otro umbral establecido.

Ambos algoritmos tienen el problema de la inicialización de los vectores de referencia, y del valor inicial (también final, en el caso del k-medias) del



número de clústers. la existencia de diferentes parámetros en el caso del ISODATA lo hace también complicado de manejar.



Los algoritmos de clustering también se pueden usar para cuantización vectorial o VQ (vector quantization), pues se pueden usar para sustituir un elemento perteneciente a un grupo por un representante de ese grupo, habitualmente llamado vector código o centro del grupo. La cuantización vectorial pretende reducir la cantidad de bits a transmitir en una línea de comunicaciones, al sustituir un vector por un código que indica qué representante se está transmitiendo (suponiendo, claro está, que ambos tengan los mismos vectores código); la técnica, además, suprime el ruido.

En todo caso, ambos son algoritmos de búsqueda, o de optimización: consiste en minimizar la distancia cuadrática media de los vectores muestra a los de referencia. ¿A alguien de los que ha seguido este tutorial se le ocurriría una manera mejor de hacerlo?

El SOM también es un algoritmo de clustering, aunque tiene un cierto valor añadido: hace unas cuantas más cosas.

Ejercicios 1
1. Ejecutar la demo online del k-medias, y ver su funcionamiento.
2. Consultar la librería en Perl para clustering, instalarla, y ejecutar algún ejemplo. ¿Qué algoritmos de clustering incluye?.

4 Métodos de proyección multidimensional

Como al mapa de Kohonen no le falta más que el volante forrado de leopardo, también puede usarse para proyección multidimensional, como hemos visto que sucede con los algoritmos de aprendizaje no supervisado. En realidad, es un método de proyección no lineal. Los métodos de proyección lineales, tales como el análisis de componentes principales, tratan de hayar un subespacio de pocas dimensiones que contenga la máxima varianza de la muestra de entrada; una vez hayado, se pueden proyectar las muestras sobre ese espacio, y usarlas para clasificación supervisada. Sin embargo, no siempre se puede proyectar linealmente a unas pocas dimensiones, y hay que usar sistemad de proyección no lineal. Uno de ellos es el denominado escalado multidimensioal (Multidimensional scaling, MDS), que usa las distancias entre los diferentes elementos de la muestra y trata de proyectarlos a un espacio de dimensión inferior, habitualmente dos. Este algoritmo trata de mantener las relaciones topológicas, igual que lo hace el algoritmo denominado mapa de Sammon, un algoritmo que proyecta a dos dimensiones un conjunto de datos n-dimensional, manteniendo las relaciones métricas (es decir, lo que esté cerca en n-dimensiones debe estar también cerca en 2 dimensiones.

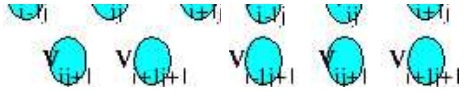
Últimamente han surgido algoritmos más potentes, como el análisis de componentes curvilíneos, desarrollado a partir del mapa de Kohonen, y que mejora el mapa de Sammon en el sentido que favorece la reproducción de distancias cortas en la proyección con respecto a las distancias largas.

5 Mapa de Kohonen: teoría

Contenido de esta sección

- Introducción teórica
- Implementación

Un mapa de Kohonen es un conjunto de vectores de dimensión n distribuidos habitualmente en una retícula bidimensional (aunque nada impide que se distribuyan en 3 o más dimensiones). Para cada vector (al que a veces llamaremos nodo o neurona) se define un vecindario: cada vector puede tener 8 vecinos (es decir, una retícula rectangular) o 6 (retícula hexagonal). Hay razones teóricas para preferir una u otra; mientras que la más popular es la rectangular, la hexagonal tiene una base teórica más sólida.



La vecindad tiene otro parámetro: la forma de la función por la que se cambian los valores de los vectores que hay en ella: una forma tipo gaussiana (gaussian) hara que el cambio de valores disminuya con la distancia, mientras que una forma tipo burbuja (bubble) cambiará de la misma forma todos los vectores que pertenezcan a la vecindad.

Para entrenar un mapa de Kohonen, primero hay que tener un conjunto de datos, que se dividirá en tres conjuntos: entrenamiento, prueba y validación. El de entrenamiento servirá precisamente para eso, el de prueba para seleccionar un mapa entre varios, y el de de validación, posteriormente, para establecer el error final, que es el que vale. Esta es solo una de las formas posibles de entranar un mapa. Para establecer el error total de un mapa con unos parámetros determinados, se puede usar otra metodología diferente llamada dejar-k-fuera (leave-k-out): se divide el connunto de datos en k partes, se usan k-1 para entrenamiento, y una para prueba; el procedimiento se repite para las k partes en las que se ha dividido el conjunto de entrenamiento.

Rara vez los conjuntos sirven tal como los dan; suelen tener la mala costumbre de tener valores en escalas muy diferentes, variabilidades muy diferentes, e incluso algunos tienen la mala costumbre de ser categóricos. Por eso, normalmente, previamente al entrenamiento, hay que realizar algún tipo de preprocesamiento, para que todas las variables tengan aproximadamente el mismo rango y la misma desviación estándar. La forma más segura es hacer lo siguiente:

- Convertir variables categóricas de n categorías en n variables (también se pueden convertir en n-1 variables). Los valores de estas variables dependerán de cómo hayamos preprocesado el resto del fichero: normalmente se pondrá al valor máximo en caso de que la variable corresponda a esa categoría y en el valor mínimo en el caso de que no. Por ejemplo, en una variable categórica con 3 categorías, la segunda categoría se codificaría como [min, max, min], que podría ser [-1, 1, -1].
- Normalizar el resto de las columnas, restando la media y dividiento por la desviación estándar. Así todas las variables tendrán un rango entre -1 y 1.

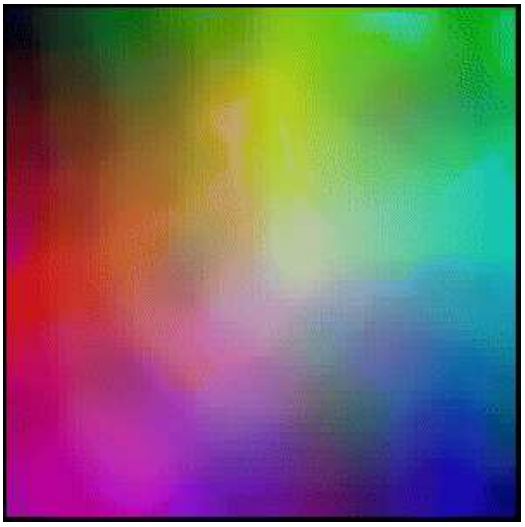
Otros preprocesos posibles son aplicar logaritmos, en caso de que el rango de variación pase por varios órdenes de magnitud, o restar el mínimo y dividir por el rango de variación, para dar diferentes variables en el rango [0,1]. Aún así, cuando se trata de variables muy diferentes, sobre todo en el caso de que la distribución de las variables sea muy diferente, los modelos que se van a hallar no son excesivamente buenos, y habría que someterlos a algún preproceso adicional.

Luego, dependiendo de la aplicación, se hacen más análisis sobre los resultados. Una de las herramientas de análisis que se puede aplicar se denomina mapa de Sammon, una herramienta de proyección bidimensional que permite proyectar un conjunto de muchas dimensiones en dos dimensiones. La visualización de proyección permite descubrir agrupamientos "naturales" del conjunto de datos; es una alternativa al mapa de Kohonen. También se suele usar para proyectar el mapa de Kohonen en sí, dándote una idea de cuáles son los clusters que forman los vectores que a su vez forman el mapa de Kohonen. En el paquete SOM_PAK se incluye `sammon`, un programa que permite calcular el mapa de Sammon. Con lo que hay que tener más cuidado es con el runlength, el número de iteraciones del algoritmo. Si se pasa uno quedará un mapilla bastante mísero. El número correcto dependerá del problema, pero un número adecuado puede ser 10. Uno de los principales problemas del mapa de Sammon es que es muy sensible a los datos que trillan fuera de parva, así que habrá que eliminarlos uno por uno antes de tener algo decente.

Ejercicios 2
1. Instalar algún paquete que incluya el mapa autoorganizativo de Kohonen, como por ejemplo, el SOM_PAK.
2. Crear en Perl un conjunto aleatorio, con 10 variables, una de las cuales es categórica con 3 categorías, otra son números reales entre 0 y 10000, y el resto son números aleatorios entre -10 y 10. Crear un programa para preprocesar este conjunto.
3. Aplicar el mapa de sammon sobre el conjunto anterior, para ver su proyección. Aplicarlo también sobre un conjunto de datos ya estructurado (el incluido como ejemplo en el paquete SOM_PAK puede servir), para apreciar visualmente el agrupamiento logrado por la proyección.
4. Usar de alguno de los sitios on-line que aplican el mapa de Kohonen: GEPAS o SOMCD. En el primero, usar el conjunto de datos creado anteriormente para ver resultados de clustering. En el segundo, usar el ejemplo para calcular los valores de estructura secundaria de una proteina a partir de los datos de dicroismo circular ejemplo u obtenidos de otro sitio.

Una vez el conjunto está bien preprocesadito, en un formato que sea el adecuado para pasárselo al programa correspondiente, viene el entrenamiento en sí, y la arquitectura de la red elegida (hexagonal o cuadrada). En dos palabras, consiste en lo siguiente:

- Inicialización de los vectores del mapa de Kohonen.. Se puede hacer de muchas formas diferentes: aleatoriamente dentro del rango de variación de cada variable, y con una distribución determinada: uniforme, gaussiana, o bien aleatoriamente con valores pequeños, mucho menores que ese rango de variación; usando vectores del conjunto de entrenamiento, lo cual hace que no tenga uno que preocuparse por el rango de variación, o bien usando algún algorimo tal como el k-medias, que permite tener ya inicialmente una población de vectores que cubre más o menos el espacio de entrada.
- Entrenamiento:
  - se escoge aleatoriamente un vector del conjunto.
  - se calcula el vector más cercano de entre los del mapa, y se le denomina ganador;
  - se cambia el valor de ese vector y de otros vectores en su vecindad de forma que se acerquen más al vector de entrada. El tamaño de la vecindad disminuirá a lo largo del entrenamiento; esa es la clave de la autoorganización.



Normalmente el entrenamiento se divide en dos fases: una primera en la que la vecindad disminuye, y cuya longitud viene a ser unas 10 veces el tamaño del conjunto de entrenamiento, y una segunda en la que la vecindad permanece fija (a veces llamada fase de autoorganización), que suele durar 100 veces el tamaño del conjunto de entrenamiento.

La tasa de cambio de los vectores ganadores y su vecindad también varía durante el entrenamiento; tiene al principio un valor, y va disminuyendo hasta ser cero al final de la fase. Normalmente, para la primera fase se suele usar una tasa del orden de 0.1, y para la segunda, valores bastante menores (la décima parte, aproximadamente; aunque todas estas cantidades son recetillas de usuario habitual que pueden variar ligeramente en algún problema en particular).

El entrenamiento consiste, evidentemente, en un procedimiento iterativo de minimización del error cuadrático medio; por lo tanto, cuanto menor sea este error, mejor. Sin embargo, tampoco conviene que sea demasiado pequeño, porque se trataría de sobreentrenamiento. Como regla para evitar el sobreentrenamiento, el modelo (en este caso el mapa de Kohonen) no debe tener más de 0.1 veces el número de parámetros en el conjunto de entrenamiento, es decir, no superar el 10% de su tamaño. En algunas aplicaciones, sin embargo, el mapa de Kohonen puede ser incluso mayor que el conjunto de entrenamiento.

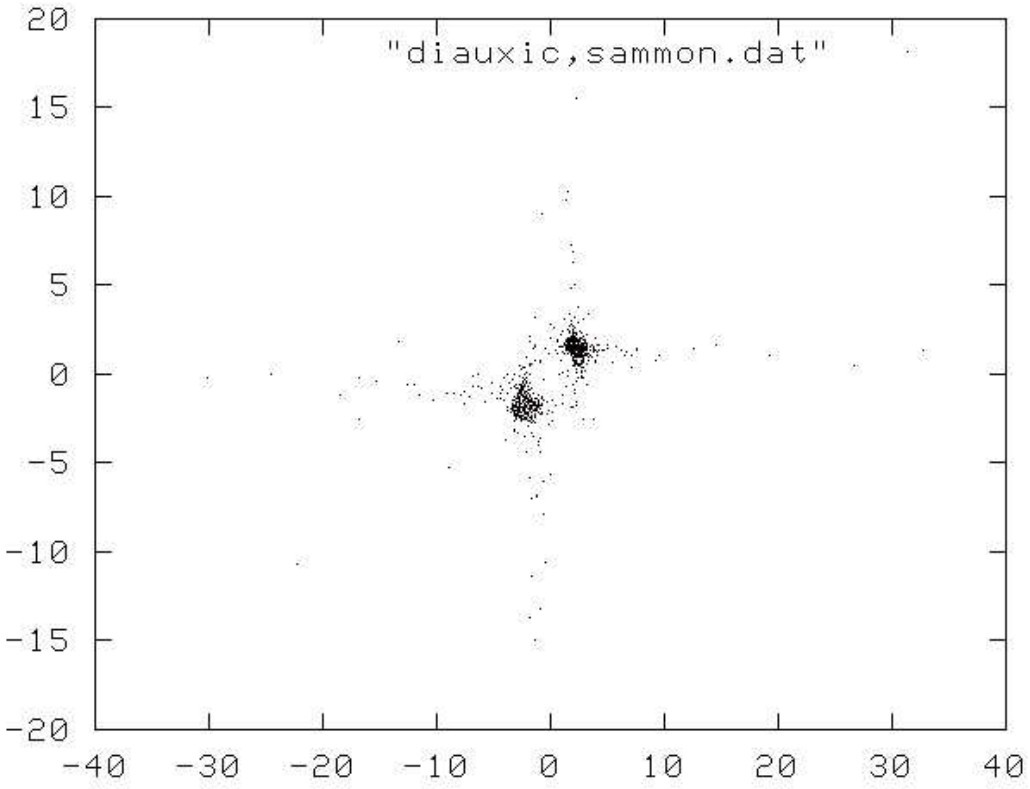
Ejercicios 3
1. Usar SOM_PAK sobre el conjunto de entrenamiento aleatorio creado anteriormente. Inicialmente usar el comando `vfind`, que va preguntando los diferentes valores. Posteriormente, crear un fichero de parámetros, y usadlo de la forma siguiente: `vfind < fichero.param`.
2. Tomando alguno de los conjuntos de datos de GEPAS, aplicadle el mapa de Kohonen usando SOM_PAK. Estimad cuál es la mejor mezcla de parámetros usando como medida el error cuadrático medio.

6 Mapa de Kohonen: práctica

Contenido de esta sección

- Calibrando mapas
- Visualización

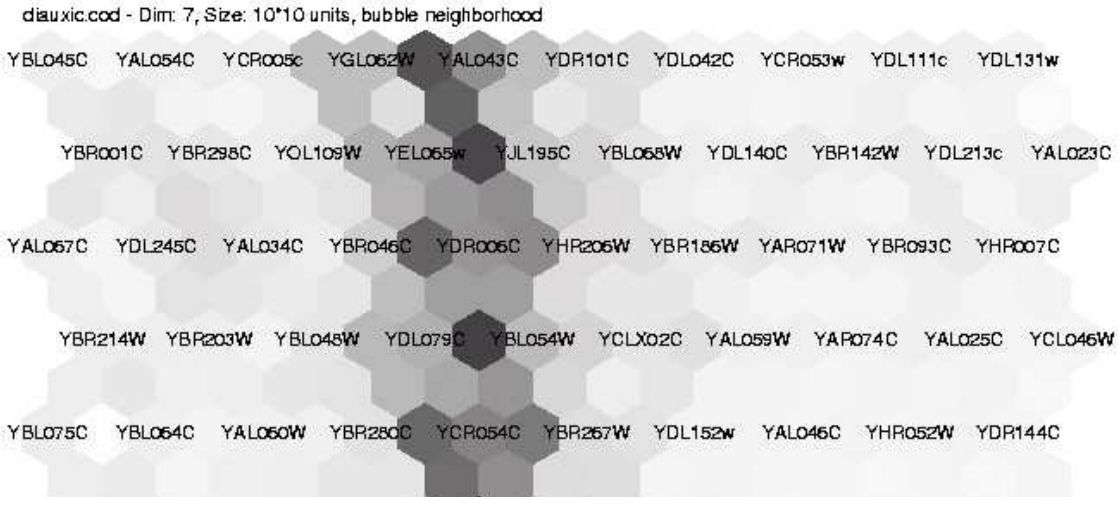
Una vez entrenado el mapa, se obtiene un fichero de pesos, un conjunto de vectores que constituyen el mapa entrenado. ¿qué se puede hacer con él? Pues usarlo para algo, se supone. A pesar de que el algoritmo de Kohonen es no supervisado, en la mayor parte de los casos se usan etiquetas para identificar de alguna forma cada uno de los datos que se le ha pasado. Esas etiquetas se van a visualizar sobre el mapa, para ver dónde ha colocado el entrenamiento los datos con diferentes características. Lo primero es calibrar el mapa, asignando a cada vector la etiqueta del más cercano de entre los que hay en el conjunto de entrenamiento. Para calibrar se usa, dentro del paquete SOM_PAK, el programa `vcal`.

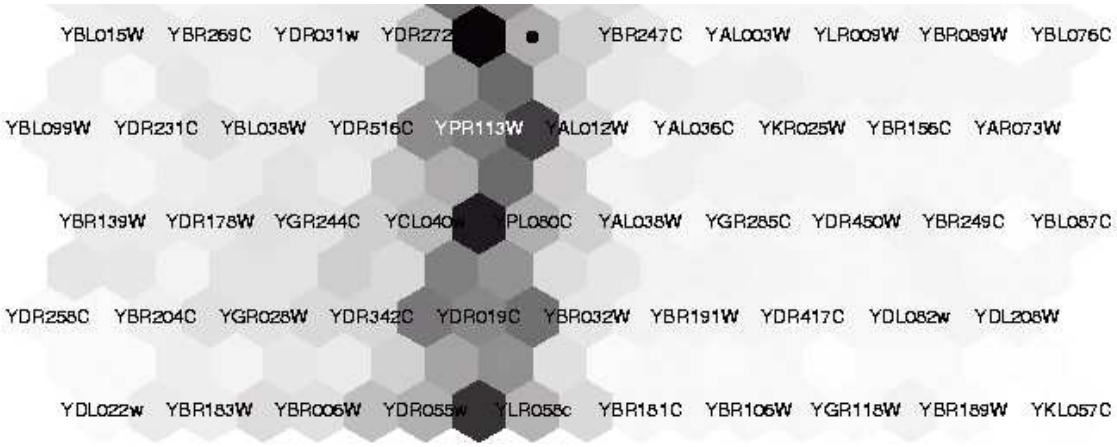


Ejercicios 4
1. Aplicar el mapa de Sammon a los datos del Diauxic shift, tratando los outliers adecuadamente. Describir qué estructura tienen los datos. Aplicar posteriormente el mapa de Kohonen, y ver qué grupos se forman.

Otra herramienta que permite apreciar el agrupamiento de los datos es el algoritmo UMatrix, que se usa sobre el mapa ya calibrado, y da una medida de la distancia entre dos elementos del mapa, y la distancia media de cada elemento a los que lo rodean. Esto permite apreciar los clusters: son zonas "claras" separadas por "zonas oscuras". Es una forma un tanto visual de verlo, pero es tan objetivo como asignar un umbral determinado, y decir que toda distancia por encima de ese umbral hace que se separen los clusters.

UMatrix se aplica en SOM_PAK usando el programa `umat`, y, aplicándolo sobre un mapa entrenado con los datos del Diauxic Shift usados anteriormente, da un resultado como el que se muestra en la imagen

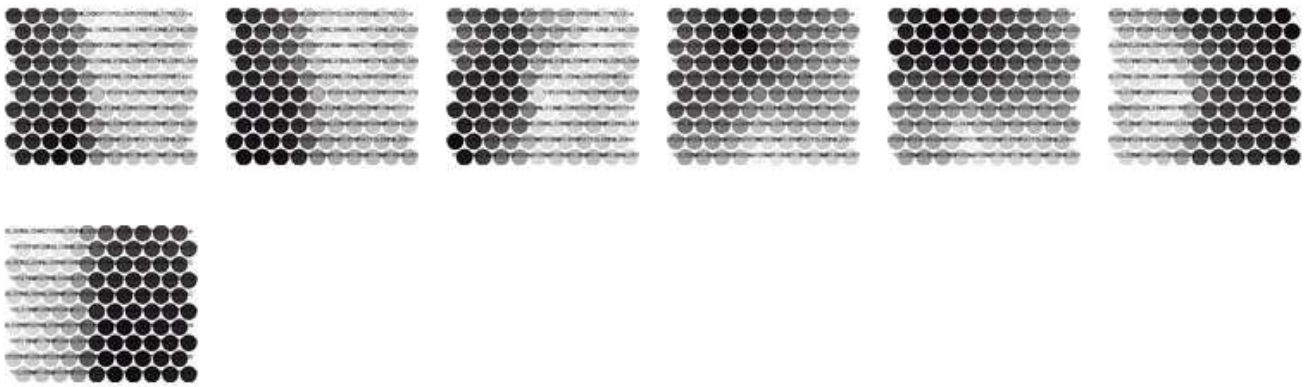




En la imagen se ve, por ejemplo, que hay dos clusters, que se reparten más o menos a partes iguales el espacio en el mapa de Kohonen; están separados por una banda central donde la distancia entre vectores es mayor. En este caso, el mapa de Kohonen refleja la misma estructura de dos clusters que se veían en el mapa de Sammon; aunque no siempre están las cosas tan claritas.

Ejercicios 5
1. Comparar los resultados obtenidos observando el UMatrix con los resultados de los algoritmos SOTA y SOMTree tal como aparecen en el GEPAS. ¿Cuántos clusters hay? ¿Cuántos se aprecian?

La última fase de análisis de datos usando el mapa de Kohonen consiste en el análisis de planos, es decir, ver cómo se distribuyen los valores de cada uno de los componentes de los vectores del mapa a lo largo y ancho del mismo. Esta distribución permite dar reglillas sobre cuáles son los componentes más representativos en cada uno de los clusters, y qué valores de cada componente son los que los caracterizan. Los valores de los componentes se suelen representar sobre el propio mapa, con un tono oscuro indicando valores altos y un tono claro valores bajos, todo ello relativamente a los valores máximos y mínimos para cada componente.



En este caso, por ejemplo, se puede ver que el cluster a la izquierda del mapa corresonde a valores bajos de los tres primeros componentes, y altos de los tres últimos, y el cluster a la derecha justo al contrario.

Ejercicios 6
1. Sobre los datos de esporulación del GEPAS aplicar el mapa de Kohonen, extraer clusters usando el algoritmo UMatrix y finalmente, hacer un análisis por planos para caracterizar cada uno de los clusters.

Contenido de esta

7 Usando el mapa de Kohonen desde Perl

Para integrar el mapa de Kohonen en una aplicación, se puede usar SOM_PAK como programa externo, pero siempre es más fácil disponer de un módulo accesible desde un lenguaje de programación tal como Perl. Perl, además, tiene una serie de módulos, agrupados dentro del espacio de nombres `Bio`, que sirven para manejar datos relacionados con biocomputación y acceder a bases de datos relacionadas con la misma. Estos módulos están accesibles desde CPAN, el sitio de los módulos en Perl, o bien desde el sitio de BioPerl.

La serie de módulos `Bio::DB` sirve, por ejemplo, para acceder a bases de datos biológicas tales como GeneBank o SwissProt:

```
$seq_object =
      get_sequence('swissprot',"ROA1_HUMAN");
```

Estas secuencias se pueden manejar en una variedad de formatos diferentes, o realizarse una serie de operaciones sobre ellas: invertir, transformar de aminoácidos a proteinas y viceversa, o enviarse a una serie de sitios web tales como BLAST.

Finalmente, se pueden emplear para tratarlos con el mapa autoorganizativo de Kohonen, usando el módulo AI::NeuralNet::Kohonen:

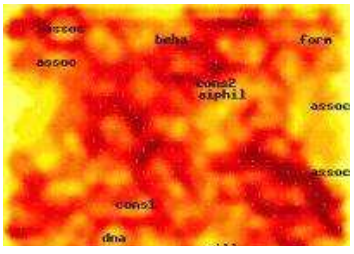
```
        $_ = AI::NeuralNet::Kohonen->new(
                map_dim_x => 39,
                map_dim_y => 19,
                epochs    => 100,
                table     =>
        "R G B
        1 0 0
        0 1 0
        0 0 1
        1 1 0
        1 0 1
        0 1 1
        1 1 1
        ");

        $_->train;
        $_->save_file('mydata.txt');
        exit;
```

Para tratar con secuencias biológicas, habría primero que tratarla para poderla meter en el argumento `table` de la red de Kohonen. El formato de salida es compatible con el mapa autoorganizativo de Kohonen; y se puede usar también para aplicar los resultados una vez entrenada la red.

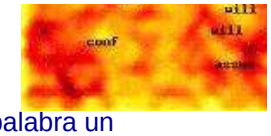
8 Aplicaciones del mapa de Kohonen: clasificación de documentos

Una de las posibles aplicaciones del mapa de Kohonen es a la clasificación de conjuntos de documentos. Genéricamente, a esta aplicación se le llama WEBSOM. Lo que se persigue con esta aplicación no es solo la clasificación de documentos, sino también el crear un mapa de un corpus de documentos, de forma que se pueda navegar por él entre documentos similares, y, además, ver la estructura a gran escala del grupo de documentos, con el objeto de detectar grupos naturales (los clusters de los que hemos hablado anteriormente).



El algoritmo funciona de la forma siguiente: primero, se crea un mapa de palabras; el objetivo es desambiguar los significados de las palabras, detectando sinónimos y otras relaciones semánticas (hiperónimos, por ejemplo). Para ello, se entrena un mapa de Kohonen con cada palabra representada por su contexto promediado: se toma cada palabra y se le asigna un vector n- dimensional aleatorio. Una vez hecho eso, se calcula para cada palabra la que le antecede y sucede, teniendo en cuenta finales de frase y principios como si fuera otra palabra. Una vez tomadas todos los antecesores y sucesores, se promedian; la representación de la palabra

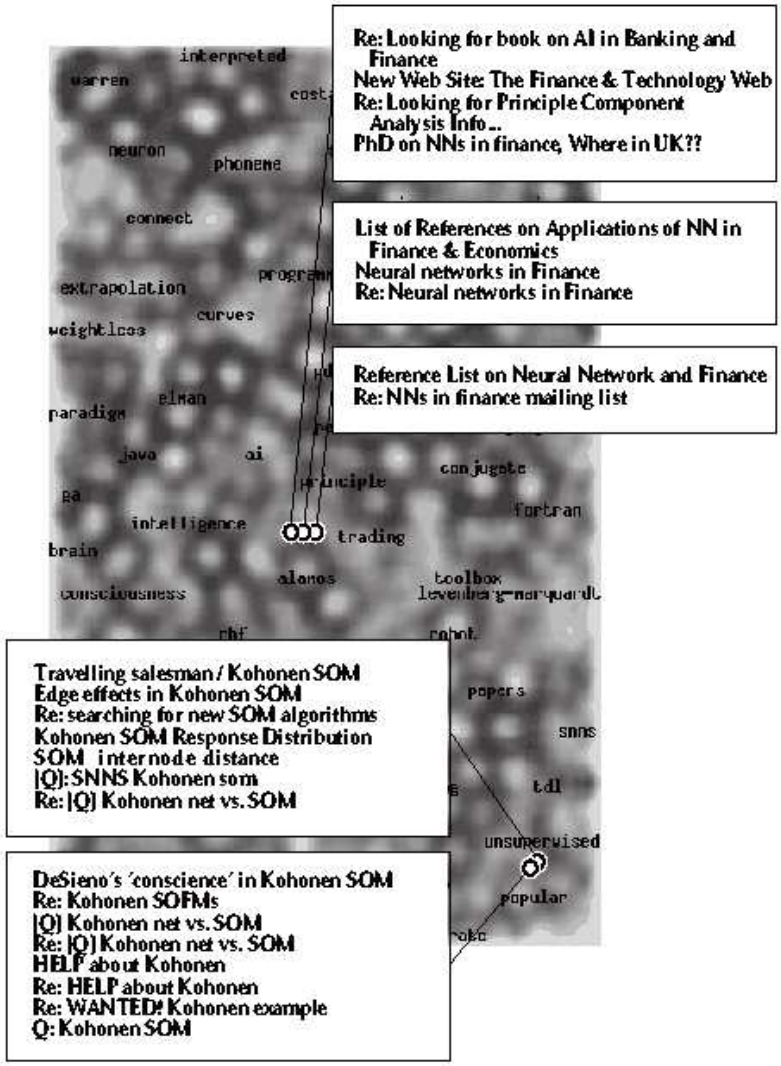
que se usará será los vectores antecesores y sucesores promedio yuxtapuestos.

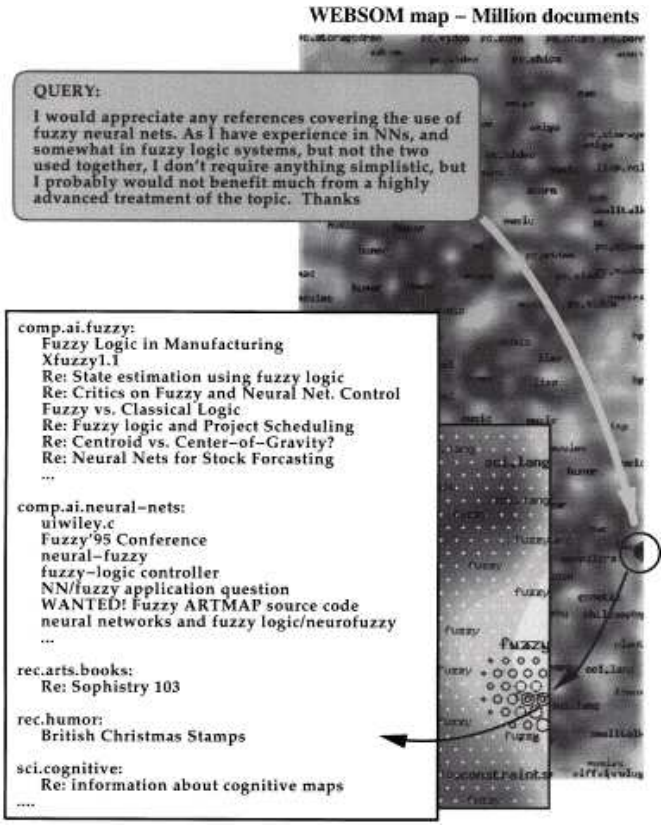


Cualquier otro método que sirva para asignar un vector único dependiente del contexto a cada palabra sirve también. Por ejemplo, si tenemos los textos categorizados (por ejemplo, secciones de un periódico), se pueden usar para cada palabra un vector de las frecuencias con las que aparece en cada sección. También se pueden considerar contextos más extensos: de 4 palabras, en vez de dos palabras. O se puede representar cada palabra con un vector que represente co-ocurrencia de una palabra en un documento determinado; esta codificación se usa para el análisis semántico latente. Dependiendo de la aplicación, será necesario uno u otro.

El resultado de entrenar un mapa de Kohonen con este conjunto de palabras es un mapa de palabras, que se supone que van a estar organizadas por significado (en realidad, por contexto): palabras similares, incluso sinónimos, irán a parar al mismo nodo de la red de Kohonen, y palabras que se usen en el mismo dominio irán a parar al mismo nodo o a nodos cercanos.

Este mapa se usará como base para la codificación de cada documento: un documento será un histograma formado por el número de veces que se "activa" cada nodo del mapa de palabras; es decir, cada documento irá representado por un vector que contendrá como componentes la cantidad de veces que se haya activado cada nodo del mapa de palabras de Kohonen. A su vez, estos vectores se usarán para entrenar un nuevo mapa de Kohonen, que será, en este caso, un mapa de documentos. Los resultados son subjetivamente buenos, permitiendo, además, navegar sobre colecciones de documentos por similaridad entre los mismos, algo que no permiten otras técnicas de codificación de los mismos. Hay una demo online de los resultados usando grupos de noticias de Usenet.

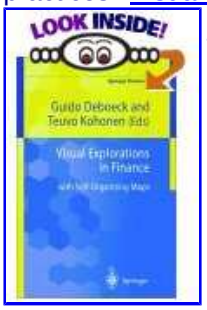




Adicionalmente, un grupo de Vigo presenta una aplicación del mapa de Kohonen para clustering y visualización de documentos.

9 Bibliografía y enlaces

La red de Kohonen viene bien explicada en su propio libro, Self-Organizing Maps, de Teuvo Kohonen, aunque hace mucho énfasis en aspectos teóricos y poco en los prácticos. Visual Explorations in Finance: With Self-Organizing Maps (Springer Finance) by Guido J. Deboeck (Editor), Teuvo Kohonen (Editor) tiene un enfoque mucho más práctico, aunque está principalmente enfocado a información financiera.



Por último, un tutorial en castellano sobre uso de la red de Kohonen: uno en la Universidad de Cádiz , parte de una presentación; una pequeña introducción al mismo, parte de un curso sobre RNs para GNU/linux, y