VNIVERSIDAD NACIONAL
AVTÓNOMA DE
MEXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

DISCRIMINACIÓN DE VARIABLES PARA MAPAS AUTOORGANIZADOS: UN ESTUDIO EN SECUENCIAS DE DNA

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN CIENCIAS (COMPUTACIÓN)

PRESENTA:
FELIPE DE JESÚS NAVARRETE CÓRDOVA

TUTOR
PEDRO MIRAMONTES VIDAL
FACULTAD DE CIENCIAS
UNAM

MÉXICO, D. F. JUNIO 2015

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

En diferentes áreas del conocimiento humano, los datos, sujetos de un estudio, son obtenidos mediante diferentes mediciones a diversos objetos que pueden ser caracterizados por una colección de variables. La relación que guardan entre sí se desconoce y a menudo, el análisis de la información generada a partir de éstos es indispensable para la toma de decisiones o entender mejor las características entre los objetos a examinar.

El análisis a realizar pretende encontrar, en muchas ocasiones, la similitud entre cada objeto, así como su clasificación, por lo que el uso de algoritmos de agrupación (o *clustering*), clasificación, etcétera, son aplicados a las colecciones de datos con los que se quiere trabajar a fin de encontrar características semejantes que permitan entender más a fondo la estructura intrínseca de cada objeto y facilitar el estudio de lo que el investigador está desarrollando.

Al abordar un problema nuevo a través de sus datos, el desconocimiento del mismo propicia al investigador medir elementos que cree que son importantes para el mismo mediante diferentes variables. Sin embargo, no todas son importantes ni brindan información útil. Es por este motivo que técnicas dentro de la minería de datos, como la selección de variables, se utilizan para reducir el número de ellas, manteniendo las más relevantes para el problema y así, poder trabajar con menor cantidad de medidas que estadísticamente, representen la misma muestra, para poder aplicar después los algoritmos de clasificación o agrupación.

Los métodos de selección, agrupamiento y clasificación deben ser capaces de relacionar a los datos para poder explorar el espacio muestral y de esta manera generar resultados que permitan inducir conclusiones al investigador.

Esta tesis busca observar las agrupaciones obtenidas de un conjunto de datos al aplicar selección de variables y al no aplicarla, para comprobar que efectivamente, la selección de variables juega un rol importante para el análisis. Dentro de un mundo de diferentes métodos utilizados en estos dos terrenos (selección y agrupación), el trabajo se limita a la utilización de los mapas de Kohonen (SOM) tanto para la selección, como para la clasificación. La justificación del uso de es-

te algoritmo es porque permite encontrar relaciones altamente no lineales entre los datos y porque la estructura de esta red neuronal permite ver las relaciones guardadas entre los objetos de forma visual.

El caso de estudio de la tesis es el tratamiento de cadenas de DNA. Éstas contienen la información genética de todo ser vivo así como la herencia funcional y morfológica que una célula transmite a sus descendientes [45]. Además, nos brinda una huella o identificador único puesto que no hay dos individuos que compartan exactamente la misma secuencia genómica [29]. Cada cadena de DNA es codificada generando vectores reales que contienen diferentes medidas y a priori, se desconoce una distribución estadística de datos, ni sus similitudes, por lo que la aplicación del mapa autoorganizado tanto para selección como agrupación, es *ad hoc* para tratar el problema.

En el Capítulo 2 se explica la metodología de los mapas autoorganizados así como una introducción al significado del aprendizaje artificial. En el Capítulo 3 se introduce bevemente la selección de variables, explicando algunos métodos existentes. El Capítulo 4 contextualiza el trabajo sobre algunos aspectos básicos de las cadenas de DNA y la forma en que estas se codifican para obtener los vectores reales con los que se desea trabajar. La metodología de análisis, las simulaciones y los resultados obtenidos se muestran en el Capítulo 5 y el Capítulo 6 indica las conclusiones y el desarrollo a futuro de esta tesis.

Capítulo 2

Mapas autoorganizados

Los mapas autoorganizados[1] son un tipo de redes neuronales cuyo aprendizaje es competitivo y se realiza de manera no supervisada [34, 35, 26, 6] en donde sólo una neurona es activada en determinado tiempo [34, 35, 26] de acuerdo al dato de entrada.

A diferencia de otras redes neuronales que tienen una capa de entrada, una o más escondidas y una de salida, las redes de Kohonen únicamente constan de una capa, la malla de neuronas[2], donde cada una de ellas están conectadas con un elemento de entrada mediante vectores de pesos, actualizados constantemente respondiendo a los estímulos provenientes de él.

Cada neurona tiene una posición determinada en la red y responde a un determinado patrón en el dominio de entrada [34], por lo que las neuronas se van especializando a diferentes patrones o clases de entradas a lo largo del entrenamiento. Esto tiene como consecuencia que neuronas físicamente cerca de otras respondan a clases de entrada parecidos creando un sistema de coordenadas ordenado (mapa topológico) para cada entrada sobre la malla [34, 26]. De esta manera, los datos de entrada son mapeados y ordenados a un mapa de menor dimensión [6, 16, 26] usualmente bidimensional o tridimensional. Mapas de mayor dimensión no suelen ser utilizados debido a su dificultad para la visualización [26].

El entrenamiento de estas redes es hecho en tres procesos principales, que se explican más adelante en este capítulo: competencia, cooperación y adaptación.

La utilización de mapas de Kohonen es aplicable en áreas como robótica, reconocimiento de patrones, control de procesamiento, entre otras [34] principalmente para la reducción de dimensión y agrupación de características (*clustering*).

[1] En la literatura se puede en encontrar también como mapas o redes de Kohonen o SOM, por sus siglas en inglés (*Self-Organizing Maps*).

[2] También llamadas unidades competitivas o unidades de Kohonen.

2.1. Fundamentos de los mapas autoorganizados

Funciones como el habla, visión, control de movimiento, etcétera, se realizan dentro de áreas particulares en el cerebro humano. Estas áreas contienen estructuras que representan un mapeo interno de respuestas de órganos sensoriales [6] y tienen una topología multidimensional en donde se reciben diferentes datos de los órganos, por ejemplo, percibir la información de los ojos es obtener el color, estructuras, posición, etcétera; información que debe ser procesada por el cerebro.

Como ejemplo, las neuronas asociadas a las respuestas auditivas están organizadas en diferentes regiones auditivas en el llamado mapa de frecuencias o tonotópico en donde las neuronas vecinas responden a frecuencias de sonido similares [16].

Otro ejemplo es encontrado en la corteza visual del cerebro. Las neuronas dentro de ella decodifican y procesan las imágenes visuales y mapean esta información como una proyección bidimensional en la corteza [54]. La Figura 2.1 muestra cómo está proyectado el campo visual a la corteza visual (figura izquierda). La figura de la derecha muestra el campo visual completo. El círculo central representa el centro del mismo. También muestra cómo las señales obtenidas del centro del campo de visión son procesadas en mayor detalle y resolución que aquellas del área periférica, pues la superficie en la corteza, asignada al centro de visión, es desproporcional [54].

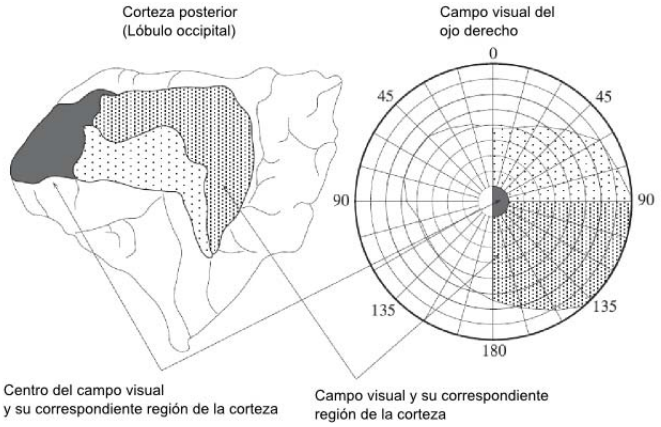


Figura 2.1: Proyección del campo visual a la corteza cerebral. La figura muestra cómo las tres diferentes áreas del campo visual son mapeadas a áreas en la corteza visual. (Imagen tomada de *Neural Networks: A Systematic Introduction* [54]).

Además de la corteza visual y las regiones tonotópicas, otros impulsos provenientes de diferentes órganos del cuerpo son relacionados a áreas dentro de la corteza cerebral en el que las neuronas responden a sensaciones similares agrupándolas en regiones contiguas de acuerdo a la similitud de ellos y generando lo que es conocido como "mapa de características ordenadas" [16].

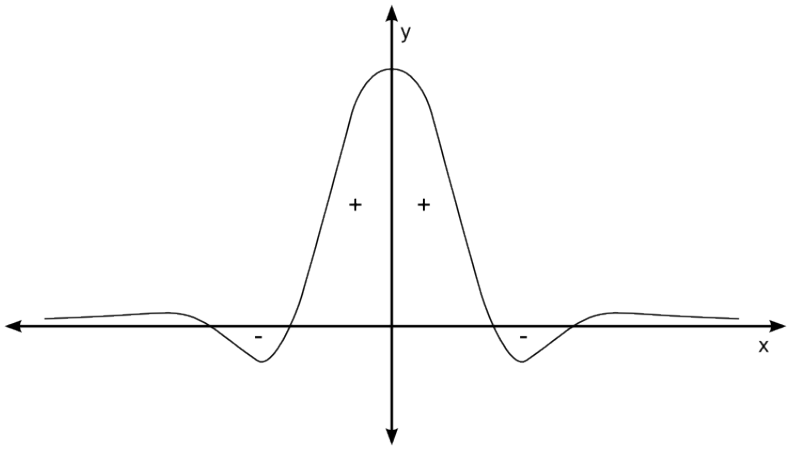


Figura 2.2: Potencial sináptico de la neurona activa. $y = 0$ representa la neurona activa. Las zonas de la gráfica con signos + indican que existe excitación por parte de la neurona activa a las neuronas dentro de esa región. El signo - indica que existe inhibición. Las neuronas más cercanas a la activa tienen mayor excitación por esta, mientras que as más lejanas tienen poca.

La activación de una neurona es propagada a otras mediante axones, que pueden causar una señal inhibitoria o excitadora para la célula nerviosa con la que hay conexión [6]. Al existir conexiones laterales entre neuronas, las células más cercanas a la activa tienen conexiones más fuertes, siendo excitadas y aquellas ubicadas a cierta distancia son inhibidas. La Figura 2.2 muestra el potencial sináptico entre las neuronas. El eje de las x indica la distancia existente entre neuronas y el eje de las y el nivel de excitación o inhibición. Las neuronas más cercanas a la activa ($y = 0$) son las que mayor excitación tienen por parte de la activa. Al incrementar la distancia a la neurona activa, las conexiones se vuelven inhibitorias y al continuar creciendo esta distancia, vuelven a ser excitadas pero cada vez con menor intensidad.

Kohonen utilizó los estudios de mapeos de la corteza cerebral para presentar los mapas autoorganizados. Las interacciones laterales entre neuronas las utilizó para modelar la adaptación de pesos a una cantidad restringida de neuronas cercana a la activa, que llamó vecindad [6].

2.2. Aprendizaje no supervisado

El aprendizaje (artificial) puede darse de dos maneras: supervisado y no supervisado. El supervisado requiere de un "maestro" que le indique al sistema (red neuronal) cómo responder a un estímulo (entrada). El maestro contiene el conocimiento del entorno mediante un conjunto de valores de entrada y salidas, por lo que conoce qué respuesta tiene cada valor de entrada [26]. Los valores de entrada son presentados a la red y la salida obtenida es medida respecto a la esperada. De esta manera, los pesos de la red son rectificados [54]. La Figura 2.3 ilustra lo anterior.

El aprendizaje supervisado puede ser categorizado en dos clases: por refuerzo y correctivo. En la primer clase sólo se sabe si la red produce la respuesta deseada

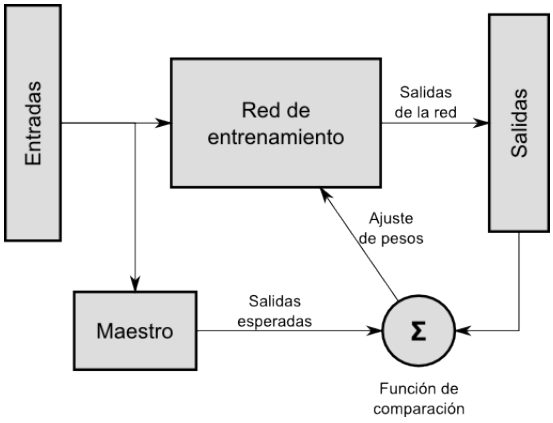


Figura 2.3: Aprendizaje supervisado. Los datos de entrada son presentados a la red para que ésta los analice y presente sus salidas. El maestro, que tiene las salidas correctas, realiza la comparación entre ambas para poder actualizar los pesos de la red.

y los pesos son actualizados basados en esa información por lo que sólo el vector de entrada puede corregirlos [54, 27] y se desconoce cuál es el valor correcto [27]. En la segunda clase, la magnitud del error junto con los vectores de entrada, determinan cómo se corrigen los pesos de la red [54] y actúa como un mecanismo de control con el fin de hacer la señal de salida lo más cercana posible a la respuesta deseada. Esta magnitud es obtenida comparando la respuesta deseada con la obtenida [26] por lo que la red sabe cuál es el valor correcto.

A diferencia del aprendizaje supervisado, el no supervisado carece de alguna guía o maestro que permita realizar correcciones al algoritmo [54]. La red debe descubrir por sí sola patrones, correlaciones, etcétera, en los datos de entrada [27].

Dentro del aprendizaje no supervisado se tienen dos clases: por refuerzo y competitivo. En el aprendizaje por refuerzo, las entradas producen refuerzos a la red de pesos con el fin de obtener un resultado deseado [54] y las salidas recurren a refuerzos positivos o negativos de acuerdo a qué tan cercanas o lejanas se encuentran del objetivo [71]. En el aprendizaje competitivo, sólo una elemento puede estar activo, por lo que cada unidad compite entre sí [54, 27].

2.2.1. Aprendizaje hebbiano

El aprendizaje hebbiano es un ejemplo de aprendizaje no supervisado por refuerzo. Dentro de este tipo de aprendizaje, la eficiencia de las sinapsis entre las neuronas se incrementa cuando dos neuronas conectadas por su sinapsis se activan simultáneamente y decrementa cuando no hay correlación de activación entre ellas [54]. A esta sinapsis se le conoce como *sinapsis hebbiana* y de acuerdo a lo anterior, incrementa su fuerza con correlación presináptica y postsináptica positiva y decrece con señales no correlacionadas o negativamente correlacionadas [26].

Las salidas de las redes basadas en este aprendizaje no tienen la idea de ga-

nador y tienen como propósito la proyección en los principales componentes de los datos de entrada en lugar de la clasificación de patrones [27].

Dentro del aprendizaje hebbiano, múltiples unidades competitivas (o neuronas) pueden estar activas a la vez [54, 27] y su expresión matemática [27, 26] está dada por

$$\Delta w_{kj}(t) = \alpha y_k(t) x_j(k) \quad (0 < \alpha < 1)$$

donde α es una constante positiva denominada tasa de aprendizaje, w_{kj} es el peso sináptico de la neurona k y x_j y y_j son señales presinápticas y postsinápticas respectivamente [26]. El cambio del peso sináptico es proporcional a la relación entre la entrada y salida.

2.2.2. Aprendizaje competitivo

Las neuronas de la red compiten entre ellas para activarse. La principal característica de las redes que utilizan este tipo de aprendizaje es la agrupación o categorización de los datos de entrada debido a que, a diferencia del aprendizaje hebbiano, sólo una neurona puede ser activada a la vez [26, 27].

De acuerdo a Rumelhart y Zipser [56] hay tres componentes básicos:

- Un conjunto de unidades idénticas salvo por un parámetro aleatorio con distribución uniforme que permita que cada unidad responda diferente con cada conjunto de datos de entrada.
- Limitar la fuerza de cada unidad.
- Permitir competencia entre las unidades de acuerdo a la respuesta obtenida para un subconjunto de datos de entrada.

Estos conceptos permiten que las unidades aprendan a especializarse en patrones similares.

En la representación más simple del aprendizaje competitivo se tiene un conjunto de entradas x_j y una capa de neuronas de salida O_i. Cada neurona está conectada con todos los datos de entrada [26, 27, 56] mediante conexiones excitadoras $w_{ij} \geq 0$. La red puede tener conexiones de retroalimentación entre las neuronas que permiten realizar conexiones laterales con las que cada neurona intenta de inhibir a cada neurona con la que está conectada.

Los pesos de cada unidad ganadora en cada paso de tiempo son atraídas en dirección a los *clusters* de los datos de entrada. El uso de pesos normalizados impide que vectores muy grandes puedan ganar la competencia muy pronto y en consecuencia, otros vectores de pesos nunca sean actualizados [54]. La actualización de pesos permite que el vector de pesos w_{ij} rote en dirección de la entrada x_j. Las actualizaciones se pueden realizar de diferentes maneras [54]:

Actualización con constante de aprendizaje en la que el peso es actualizado de acuerdo a

$$\Delta w_m = \alpha x_j$$

en donde α es una constante entre 0 y 1 y decrece a 0 mientras se va actualizando el aprendizaje de la red. Esta tasa controla las correcciones realizadas en la red permitiendo que sean más drásticas al inicio del procedimiento y se vayan haciendo más graduales.

Actualización por diferencias dada por

$$\Delta w_m = \alpha(x_j - w_m)$$

que permite que las correcciones sean proporcionales a la diferencia entre los vectores de entrada y el de pesos.

Actualización por tanda donde las actualizaciones son calculadas y acumuladas y después de un número de iteraciones las correciones de pesos son agregados a los pesos.

2.3. Mapas y algoritmo de Kohonen

Los mapas de Kohonen, a diferencia de otras redes neuronales como el perceptrón multicapa, no está organizado en múltiples capas (capas de entrada, escondidas y de salida) [6]. Se cuenta únicamente con el vector de entrada y una malla rectangular, generalmente bidimensional, donde se encuentran las neuronas de la red [26, 6] y los datos de entrada son conectados directamente con cada una de las neuronas. La retroalimentación se realiza mediante conexiones entre las neuronas vecinas de cada neurona [6]. La Figura 2.4 muestra cómo están estructuradas generalmente las redes de Kohonen.

El algoritmo puede ser resumido en tres simples reglas [26]: muestra de los datos, emparejamiento de similitudes y actualización; que son repetidos hasta que se ha completado el mapeo de características. De esta manera, las neuronas son organizadas en vecindades que fungen como clasificadores de características sobre los datos de entrada. El algoritmo es no supervisado pues no se especifica alguna respuesta para algún dato de entrada [6].

Las neuronas están conectadas a cada vector de entrada $\mathbf{x} \in \mathbb{R}^n$ y se calcula la excitación correspondiente respecto a las neuronas en la malla bidimensional mediante vectores de pesos $w_i = (w_{i1}, \ldots, w_{i_n})$, donde $i \in [1, \ldots, m]$ representa la i-ésima neurona dentro de la malla y m es el total de neuronas. La idea es que cada unidad aprenda a especializarse en diferentes regiones del espacio de entrada [54].

Cada unidad de Kohonen calcula la distancia euclidiana entre el vector de entrada $\mathbf{x}$ y su vector de pesos w_i. Otras métricas son posibles aquí como el producto

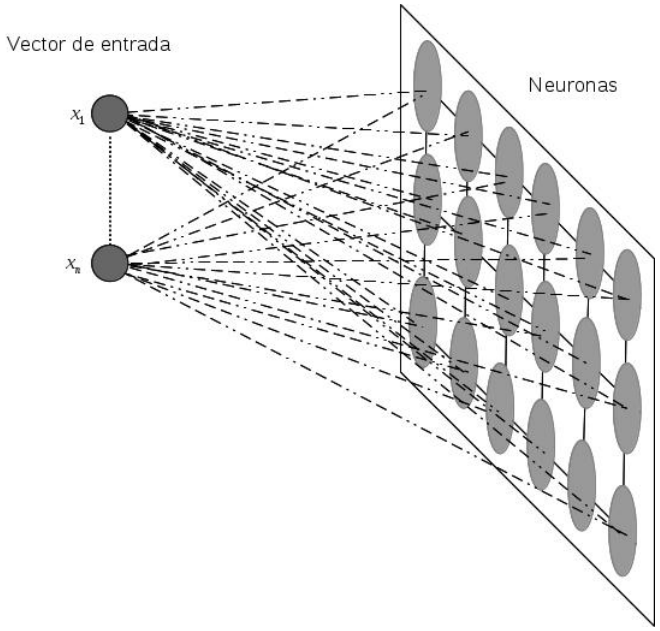


Figura 2.4: Mapa autoorganizado. Todas las entradas están conectadas a todas las neuronas.

punto $x^T w_i$. La distancia mínima define la neurona ganadora [6, 16, 34, 35, 26, 54], a la que pertenece la mayor excitación con el vector de entrada. Por otro lado, la maximización del producto punto $x^T w_i$ tiene equivalencia con la minimización de la distancia euclidiana.

El mapa autoorganizado por su inspiración en la neurobiología y la evidencia de cooperación y retroalimentación y otras interacciones laterales entre neuronas excitadas, propone que una neurona excitada de mayor excitación a neuronas cercanas a ella que a las lejanas [26, 34, 35]. Dentro de los mapas de Kohonen, estas interacciones están definidas mediante neuronas que viven en una vecindad N_k de radio r alrededor de la neurona ganadora.

Obtenida la neurona ganadora se procede a la actualización de pesos.

El Algoritmo 2.1 muestra el funcionamiento de Kohonen.

Algoritmo 2.1: Algoritmo de Kohonen.

1. Elegir valores aleatorios para los vectores $\mathbf{w}_i$.
   Definir tasa de aprendizaje α y radio r.
2. Elegir un vector de entrada $\mathbf{x} \in \mathbb{R}^n$.
3. Encontrar la unidad con mayor excitación
   $k = min_i ||\mathbf{x} - \mathbf{w}_i(t)||$ $\quad (i = 1, ..., m)$
4. Actualizar los pesos de acuerdo a la funcion de actualización:
   $w_i(t) = w_i(t) + \alpha h(i,k)(\mathbf{x} - \mathbf{w}_i(t))$ $\quad \forall i \in N_k$
5. Repetir desde 2 si el número de iteraciones no se ha alcanzado.

El paso 3 del algoritmo realiza el proceso competitivo que induce a las neuronas a competir entre ellas mediante la optimización función de discriminación (la menor distancia euclidiana de una neurona y la entrada): la neurona con el valor óptimo de la función es la ganadora. El paso 4 realiza los procesos de cooperación

y adaptación: la neurona ganadora determina la localidad espacial de una vecindad topológica de neuronas excitadas (cooperación) permitiendo incrementar sus valores mediante la actualización de sus pesos acorde al patrón de entrada al que respondió la neurona ganadora.

2.3.1. Adaptación de pesos

La actualización de pesos es proporcional a la diferencia entre el vector de entrada x_i y el vector de pesos de la neurona ganadora w_k

$$w_i(t+1) = w_i(t) + \alpha h(i,k)(\mathbf{x} - \mathbf{w}_i(t)) \tag{2.1}$$

donde i es el iésimo vector de pesos de la neurona ganadora k. La actualización no sólo cambia los pesos de la neurona ganadora, sino también de aquellas que se encuentran dentro de la vecindad N_k. Los pesos del resto de las neuronas permanecen intactos. La actualización de la neurona ganadora y de sus vecinos permite que la atracción de la neurona al vector de entrada sea para ella y sus vecinas, por lo que es deseable que el radio decrezca para reducir la influencia con otras neuronas [54]. Parte de la actualización está dada por una tasa de aprendizaje α que controla la magnitud de las actualizaciones de pesos y, de igual manera que el radio, decrece con el tiempo [54].

2.3.2. Vecindades

Como se mencionó en la Sección 2.3, existen interacciones laterales entre las neuronas excitadas y la influencia de éstas es mayor sobre neuronas cercanas. Con estas interacciones la vecindad topológica o función de vecindad, definida por $h(i,k)$, alrededor de la neurona ganadora k para un conjunto de neuronas $i \in N_k$ decrece suavemente [26]. N_k generalmente es una función monotónica decreciente, en lugar de una constante, lo que permite que la vecindad vaya disminuyendo conforme avanzan las generaciones, por lo que $N_k = N_k(t)$.

Vecindad topológica

En la literatura $h(i,k)$ tiene dos definiciones [34, 35, 54] y actúa sobre los puntos definidos en la malla de neuronas. La primera y más simple indica que

$$h(i,k) = \begin{cases} 1 & \text{si } i \in N_k \\ 0 & \text{e.o.c.} \end{cases}$$

y de esta manera la adaptación de pesos queda definida como $w_i = w_i + \alpha(\mathbf{x} - \mathbf{w}_i)$.

La otra definición, y más comúnmente utilizada, puede ser escrita en términos de una función gaussiana [34, 35, 26, 16]:

$$h(i,k) = \exp\left(-\frac{||r_k - r_i||}{2\sigma^2}\right) = \exp\left(-\frac{d(i,k)^2}{2\sigma^2}\right)$$

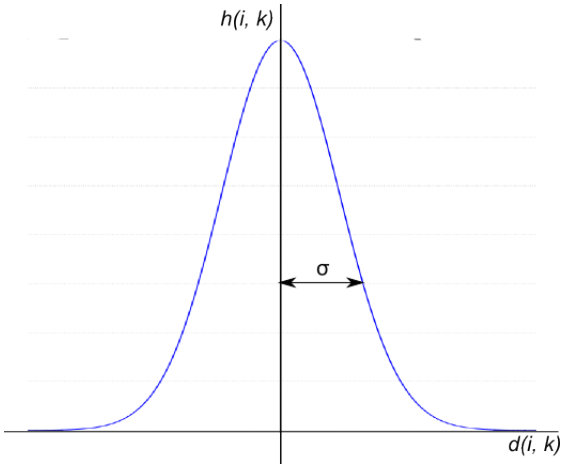


Figura 2.5: Función gaussiana para la vecindad topológica. La neurona ganadora k tiene mayor influencia ($h(i,k)$) en cada neurona i cercana (valores de $d(i,k)$ cercanos a cero) o dentro de su vecindad que aquellas que se encuentran a mayor distancia. El radio de la vecindad de la neurona ganadora está definido por σ.

donde $d(i,k)$ representa la distancia lateral [26] entre la neurona ganadora k y la neurona excitada i , mientras que el parámetro σ representa el radio de la vecindad topológica correspondiente a la función N_k [34, 35, 26].

Para la convergencia de los mapas autoorganizados se debe satisfacer

$$\lim_{t\to\infty} h(i,k) = 0$$

La función $h(i,k)$ tiene su mayor valor en la neurona ganadora k puesto que la distancia es cero y el valor de la función es $\exp(0) = 1$. Además la amplitud de la función topológica decrece monotónicamente al incrementar la distancia entre la neurona ganadora k y la neurona excitada i por lo que

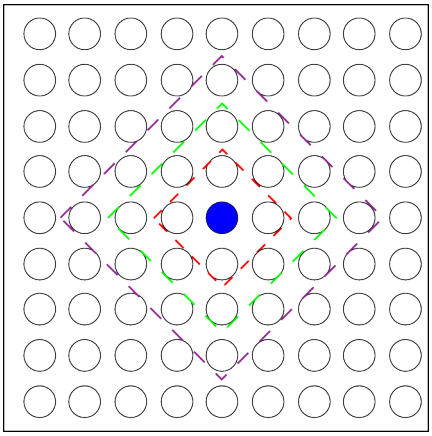
$$\lim_{d(i,k)\to\infty} h(i,k) = 0$$

Esto se puede apreciar en la Figura 2.5.

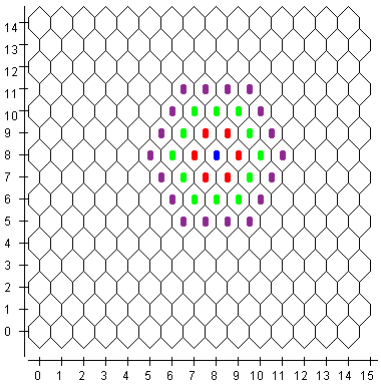
La elección de una función gaussiana para definir el potencial sináptico entre la neurona ganadora y sus vecinas está relacionada con las interacciones laterales existentes entre las neuronas del cerebro, como se vio en la Sección 2.1. La interacción cumple con una función "sombrero mexicano" que permite excitar con mayor fuerza a neuronas muy cercanas a la ganadora que a neuronas muy lejanas e inhibir a neuronas a cierta distancia. La función gaussiana elimina la inhibición y mantiene únicamente la excitación.

Reducción de la vecindad

La Figura 2.6 muestra el comportamiento de las vecindades. En una malla rectangular, las neuronas a distancia r de la neurona ganadora k son aquellas coordenadas $(k \pm i, k \pm j)$ $\forall i, j \in \{0, 1, \ldots, r\}$.



(a) Topología de la vecindad hexagonal. Cada neurona tiene seis neuronas vecinas.



(b) Topología de la vecindad rectangular. Cada neurona tiene cuatro neuronas vecinas.

Figura 2.6: La vecindad de la neurona con mayor excitación. Ésta es mostrada con el círculo azul. Sus neuronas vecinas de radio 1, 2 y 3, son delimitadas por los cuadros punteados de color rojo, verde y morado respectivamente

El tamaño de la vecindad va decreciendo con el tiempo de manera gradual, por lo que σ debe ser dependiente del tiempo y de acuerdo con Haykin [26] se puede escribir como una función exponencial decreciente:

$$\sigma(t) = \sigma_0 \exp\left(-\frac{t}{\tau_1}\right)$$

donde σ_0 es el valor de σ al inicio del algoritmo y τ_1 es una constante de tiempo, generalmente el número máximo de épocas de entrenamiento. El decrecimiento de σ permite que la función de vecindad converja a cero conforme avanza el tiempo.

Idealmente, el radio de la vecindad debe ser lo suficientemente grande. En las primeras etapas del entrenamiento, un radio amplio permite que se tenga una influencia amplia por parte de la unidad ganadora, aumentando las correcciones [54]. El decrecimiento del radio disminuye esa influencia.

2.3.3. Ordenamiento topológico

A través de la actualización de pesos definidos en la Ecuación 2.1, el vector w_k de la neurona ganadora k se mueve en dirección al vector de entrada x, así como los vectores de las neuronas vecinas. El mapa neuronal calculado por SOM es ordenado topológicamente permitiendo que la localidad espacial de una neurona corresponda a una entrada en particular [26].

Para mostrar cómo se realiza este proceso, Kohonen desarrolló una manera de visualizar la adaptación de las neuronas con las entradas. Esto lo realiza graficando los pesos de las neuronas entrada contra entrada del vector, es decir, para un vector $w_i = (w_{i1}, w_{i2})$, se grafica w_{i1} contra w_{i2} que resulta en un punto en el espacio bidimensional. Cada coordenada dada por el vector w_i se une mediante una línea con sus vecinos en el espacio físico, es decir, si w_i y w_j son vecinos en

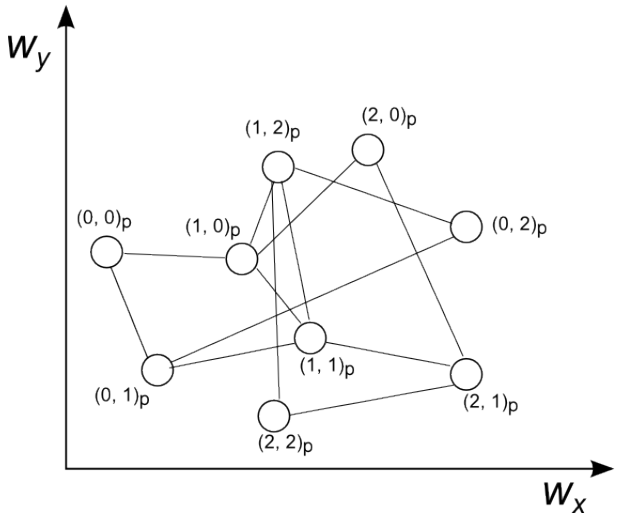


Figura 2.7: Espacios de coordenadas dentro de SOM. Cada unidad está representada por dos conjuntos de coordenadas: la coordenada del espacio físico, dentro de la malla de neuronas, dado por $(u, v)_p$ y las coordenadas definidas por el vector (w_x, w_y) dentro del espacio de pesos. La gráfica se realiza mediante las coordenadas del espacio de pesos y las líneas que unen a cada unidad son definidas por su posición en el espacio físico.

la malla de neuronas, se pinta una línea entre las dos coordenadas dadas por los vectores en el espacio de pesos. La figura 2.7 muestra lo anterior.

La evolución de la malla bajo este esquema de visualización se puede observar en la Figura 2.8 en donde al inicio, los pesos son definidos aleatoriamente y al paso de las épocas, el mapa va obteniendo la forma de una red esparciéndose para cubrir todo el espacio de entrada aproximando así la distribución, desconocida, del espacio de entrada [34, 6, 16, 26].

La prueba matemática de convergencia sólo es válida para mallas unidimensionales desarrolladas en espacios unidimensionales [62] mientras que para las de mayores dimensiones no existe una prueba general [54].

Métricas de ordenamiento

Para medir la efectividad de los mapas se han investigado algunas métricas que permiten calificar la preservación topológica de los mapas.

Una de las métricas más comunes es el error cuantitativo que mide la distancia promedio entre cada vector y la neurona con mayor excitación y es calculado mediante

$$qe = \frac{1}{N} \sum ||x_i - w_{x_i}|| \tag{2.2}$$

donde N es el número de vectores de entrenamiento y m_{x_i} es el vector de pesos correspondiente a la neurona con mayor excitación respecto al vector de entrada x_i. Con este error se puede evaluar cómo se ajusta el mapa neuronal a los datos

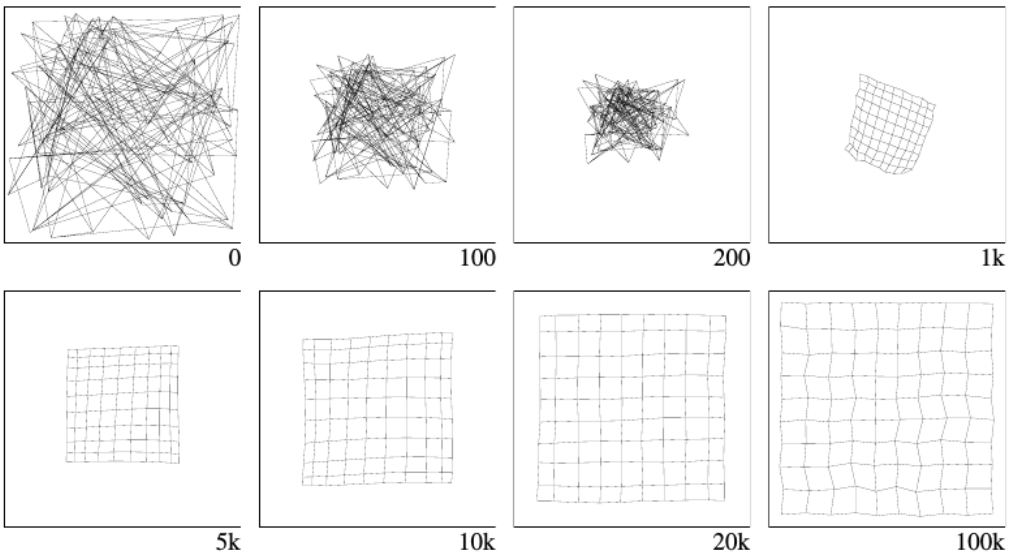


Figura 2.8: Evolución de pesos del SOM. Mediante cada iteración, se crea un ordenamiento topológico que pretende aproximar la distribución del espacio de entrada. (Imagen tomada de *Self-Organizing Map* por Van Hulle [62].)

y entre más pequeño sea el error, mejor evaluación tiene [3].

Otra alternativa para medir la preservación topológica es mediante el error topológico [32] que mide la proporción de todos los vectores de entrenamiento en los que las dos neuronas con mayor excitación no son adyacentes entre sí. Este error se calcula mediante

$$te = \frac{1}{N}\sum_{i=1}^{N} u(x_i) \tag{2.3}$$

donde la función $u(x_i)$ está definida como:

$$u(x_i) = \begin{cases} 1 & \text{si son adyacentes} \\ 0 & e.o.c. \end{cases} \tag{2.4}$$

De esta manera, entre menor es el error, se cuenta con mayor preservación topológica del mapa.

2.4. Learning Vector Quantization

Learning Vector Quantization (LVQ) es una técnica supervisada para clasificación estadística y reconocimiento de patrones [26, 6, 35, 47] cuyo propósito es representar *regiones de clases* en el espacio de entrada (vectores de entrada) mediante el aprendizaje de prototipos[3][47].

[3]En la literatura también se les conoce como vectores *codebook*.

Por ser un método no supervisado, las clases son predefinidas dentro de los datos de entrada, por lo que cada vector de entrada está ligado a una clase y cada clase puede tener diferentes vectores prototipos [27].

La idea de LVQ está basada en un método de aprendizaje no supervisado conocido como *vector quantization (VQ)* [22] en el que los vectores de entrada $x \in \mathbb{R}^n$ son categorizados en m clases para así poder representar cualquier vector por la clase en la que éste "cae". La categorización es hecha mediante vectores prototipo (pesos) $w_i \in \mathbb{R}^n$, $i = 1, 2, ..., m$ y la aproximación de x se realiza al encontrar el vector prototipo w_c más cercano a x, generalmente mediante una distancia Euclidiana [35, 27]:

$$|x - w_c| = \min_i \{|x - w_i|\} \tag{2.5}$$

Los vectores prototipo, de esta manera, dividen el espacio de entrada en un mosaico de Voronoi (*Voronoi Tesselation*) como se muestra en la Figura 2.9. Los puntos muestran los vectores prototipo y las líneas delimitan las regiones determinadas por los prototipos. Estos vectores son utilizados para reconstruir los vectores de entrada por lo que las aplicaciones de VQ son muy útiles en la compresión de datos, por ejemplo, para la transmisión y almacenamiento de datos de imágenes o voz.

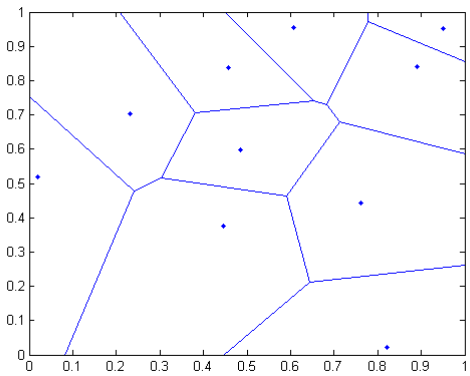


Figura 2.9: Mosaico de Voronoi (Voronoi Tesselation). Los puntos muestran los vectores prototipo y las líneas delimitan las regiones determinadas por ellos.

La supervisión del aprendizaje hecho por LVQ mediante la información de clases permite que los vectores prototipo dentro del mosaico de Voronoi se muevan ligeramente para mejorar la clasificación [26]: el vector prototipo w se mueve hacia la dirección del vector de entrada x (elegido aleatoriamente dentro del conjunto de entrada) siempre que la clase a la que pertenece x y el vector w sean acordes, y se mueve en dirección contraria en otro caso.

Dados el conjunto de vectores de entrada $\{x_i\}_{i=1}^N$, los vectores prototipo $\{w_j\}_{j=1}^m$, y $c = \min_k\{|x_i - w_k|\}$ el índice asociado al vector w_j más cercano al vector de

entrada x_i y sean C_{w_c} y C_{x_i} las clases asociadas a los vectores w_c y x_i respectivamente, el procedimiento LVQ realiza la actualización de pesos de la siguiente manera [35]:

$$w_c(t+1) = \begin{cases} w_c(t) + \alpha(t)[x_i - w_c(t)] & \text{si } C_{w_c} = C_{x_i} \\ w_c(t) - \alpha(t)[x_i - w_c(t)] & \text{si } C_{w_c} \neq C_{x_i} \\ w_c(t) & \text{e.o.c} \end{cases} \tag{2.6}$$

En la ecuación 2.6, denominada LVQ1 por Kohonen [35], α es la tasa de aprendizaje, que al igual que en los modelos de aprendizaje competitivo decrece con el tiempo. En el primer caso, cuando las clases son iguales, el vector w_c se mueve en dirección al vector de entrada mientras que en el segundo caso en dirección contraria permitiendo minimizar las clasificaciones incorrectas.

Otras propuestas han surgido y mejorado LVQ1, tal como LVQ2 [35] el cual es cercano a la teoría de clasificación Bayesiana. Las actualización de los pesos es aplicada sólo si

- el vector de entrada x_i es clasificado incorrectamente por la unidad w_c
- el siguiente vecino cercano c' tiene la clase correcta, y
- el vector x_i se encuentra muy cercano al límite de la región definida por w_c y $w_{c'}$

En este caso ambos vectores, w_c y $w_{c'}$ son actualizados de acuerdo a la regla de actualización definida por la ecuación 2.6.

La relación entre SOM, VQ y LVQ es muy cercana. SOM permite encontrar las regiones de Voronoi de manera no supervisada mediante los pesos sinápticos de las neuronas dentro de la red. Además LVQ permite que los resultados obtenidos por SOM sean mejorados al utilizar SOM y LVQ conjuntamente: el mapa entrenado de SOM es el espacio de entrada para LVQ [35, 6, 26] lo cual se ilustra en la Figura 2.10.

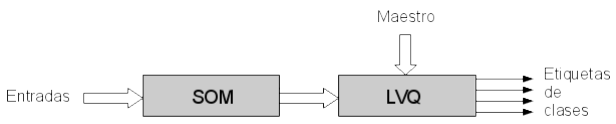


Figura 2.10: SOM y LVQ para clasificación de patrones.

2.5. Avances en los mapas autoorganizados

El algoritmo discutido en la Sección 2.3 introduce la metodología básica para mapear de manera no supervisada características similares de los vectores de entrada, sin embargo, diferentes modificaciones se han realizado al algoritmo de Kohonen. Algunas de ellas son descritas brevemente en esta sección.

2.5.1. Mapas topológicos crecientes

La malla que define el mapa de neuronas bajo este esquema de algoritmos, los *Growing Self-Organizing Maps* [4] no mantiene un tamaño ni estructura fija, sino que se va construyendo gradualmente mediante la inserción y posible eliminación de neuronas y conexiones entre ellas [62]. La idea de tener este tipo de algoritmos es para lidiar con algunos de los problemas presentes en SOM: La especificación de un tamaño adecuado del SOM y que la estructura predefinida puede no ser adecuada para representar los datos [20].

Este tipo de modelos tienen en común lo siguiente [19]:

- La red es una gráfica no dirigida en la que cada neurona es un vértice. Los vértices están conectados mediante aristas.
- Cada neurona i tiene un vector de peso w_i, n dimensional, asociado a ella.
- Los vectores de pesos de la neurona ganadora s y sus vecinas son actualizadas mediante

$$\Delta w_s = \alpha_s(v - w_s) \tag{2.7}$$
$$\Delta w_i = \alpha_i(v - w_i) \forall i \in N_s \tag{2.8}$$

  donde N_s es la vecindad topológica de la neurona ganadora s y α es la tasa de aprendizaje.

- En cada paso del entrenamiento, el error local ΔE_s de la neurona ganadora es actualizado mediante alguna métrica. Por ejemplo, para VQ, la regla de actualización está dada por la norma Euclidiana

$$\Delta E_s = ||v - w_s||^2 \tag{2.9}$$

  Este error es utilizado para determinar dónde insertar nuevas neuronas después de un número de pasos. Después de la inserción, el error es redistribuido localmente permitiendo agregar una nueva neurona en pasos subsecuentes en cualquier otra región.

Fritzke define en [17] la topología de la una nueva red construida llamada *Growing Cell Structure (GCS)* que incluye las propiedades comunes. La topología inicial es k dimensional, en la que dependiendo de k se pueden formar líneas, triángulos, tetraedros o hypertetraedros como se muestra en la Figura 2.11 . Durante el proceso de entrenamiento se agregan nuevas neuronas (o celdas, como el autor se refiere) y se eliminarán aquellas que no aporten información.

La adaptación de este método se realiza y al final de determinados λ pasos se determina la unidad q con el mayor error acumulado y la arista de mayor longitud entre el vértice q y sus vecinas es dividio para insertar ahí una nueva.

Algunas propiedades de este modelo diferentes de SOM son [20]:

[4] Se utiliza su término en inglés para facilitar el contexto con la literatura.

- generación dinámica de la estructura de la red
- adaptación constante y pequeña
- radio de la vecindad constante y pequeño

mientras que las propiedades heredadas del SOM son:

- dimensión de la red fija
- definición de la vecindad mediante la estructura topológica

Growing Neural Gas

Martinetz y Shulten [43] definen el método *Neural Gas (NG)* en donde por cada vector de entrada v se actualizan los k vecinos más cercanos de la neurona ganadora. k, al igual que la tasa de aprendizaje decrecen constantemente hasta que únicamente sea la neurona ganadora la que se actualiza. Los pesos sinápticos son adaptados independientemente de cualquier arreglo topológico en la red, es decir, el cambio de los pesos no es determinado por la disposición de las unidades dentro de la malla topológica sino por su distancia relativa entre las unidades dentro del espacio de entrada [43], por lo que la función de vecindad ahora es dentro del espacio de entrada [62]. NG no realiza inserción o eliminación de nodos. Algunos problemas que presenta esta propuesta es que se requiere *a priori* definir el número de neuronas, por lo que Fritzke propone el método *Growing Neural Gas (GNG)* [18].

Este método no da restricciones a la malla neuronal y permite su actualización mediante aprendizaje hebbiano competitivo [42] en el que se agrega una arista entre la neurona ganadora y la segunda en cada paso del entrenamiento (siempre que esta arista no exista aún) y genera una subgráfica de la triangulación de Delaunay correspondiente a los vectores prototipos.

El algoritmo inicia con dos neuronas conectadas colocadas aleatoriamente. A diferencia de NG, GNG permite la adición de nuevas neuronas: cada λ unidades de tiempo se determina la neurona q con el mayor error cumulado y una nueva neurona se agrega entre ella y uno de sus vecinos y los errores son redistribuidos

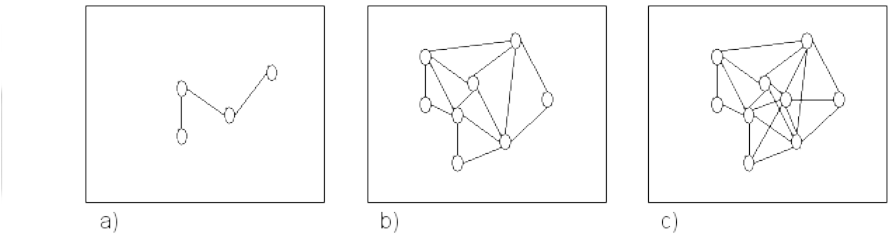


Figura 2.11: Topología de las neuronas. En a) $k = 1$. En b) $k = 2$. En c) $k = 3$.

localmente.

La topología de la red, tanto de NG como de GNG, refleja la topología del espacio de entrada y permite diferentes dimensiones en diferentes partes del espacio de entrada. Por esta razón la visualización sólo es posible para datos de dimensión pequeña [19, 62].

2.5.2. Otros avances

Similar al trabajo de Fritzke, Bauer y Villman [60] proponen un modelo incremental del SOM llamado *Growing Self-Organizing Map (GSOM)* en donde la malla de neuronas es adaptada dimensionalmente generando una estructura hiper rectangular al agregar filas o columnas completas.

Incremental Grid Growing [8], agrega nuevas neuronas en los bordes de la malla neuronal además de, agregar o eliminar conexiones basándose en las similitudes de los pesos de las neronas conectadas y similar a este esquema, otro GSOM, pero de Alahakoon [1] en el que sólo se agregan neuronas (en los bordes de la malla neuronal) y contiene un factor de esparcimiento permitiendo controlar cmo se expande la malla.

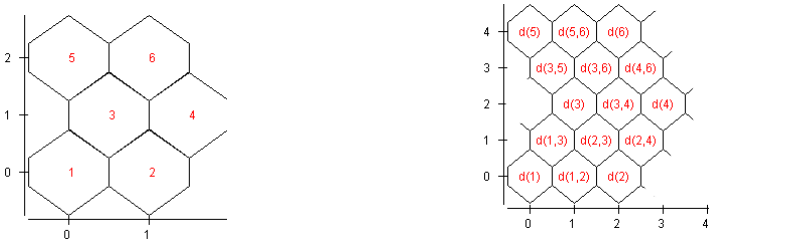
Rauber propone *Growing Hierarchical Self-Organization Maps (GHSOM)* [51] cuya arquitectura crece en el sentido jerárquico (permite descomposición jerárquica y navegación en subpartes de los datos) y horizontal (el tamaño de cada mapa individual crece acorde a los requerimientos de los datos).

Ritter propone un modelo de SOM para tratar con espacios no Euclidianos [53]: un mosaico regular del plano hiperbólico, caracterizado por una curva constante negativa gaussiana [48].

Neme [46] realiza una de las modificaciones más recientes mediante el método *Self-Organizing Maps with Selective Refractoriness (SOMSR)* en donde la vecindad es definida a partir de las unidades afectadas durante la actualización en lugar del enfoque clásico: las unidades ganadoras. De esta manera el radio de vecindad de la unidad ganadora no decrece, sino que las neuronas se convierten "insensibles" a cierta unidad ganadora [46].

En cuanto a la visualización también se han realizado avances. Por ejemplo Ultsch y Siemon proponen el método de distancias unificadas *U-Matrix (unified distance matrix)* [61] que permite visualizar los *clusters* ante un espacio de datos de gran dimensión. Mediante este método, la malla de neuronas es mapeada a otra malla que contiene las distancias entre neuronas y cuya dimensión es $2n-1 \times 2m-1$ donde n y m son el número de filas y columnas de la malla neuronal. La Figura 2.12 muestra cómo se genera la malla de distancias unificadas. En 2.12a se muestra la topología de las seis neuronas numeradas en una malla hexagonal. En la Figura 2.12b se muestra la matriz unificada, donde los valores $d(i, j)$ son las distancias

de los pesos ya entrenados entre la neurona i y la neurona j. Cada valor $d(i)$ representa el promedio aritmético de los valores que lo rodean, por ejemplo, $d(4) = \frac{d(2,4)+d(3,4)+d(4,6)}{3}$.



(a) Topología de las neuronas.

(b) Matriz de distancias unificada correspondiente a la malla de neuronas.

Figura 2.12: Matriz de distancias unificada.

2.6. Ejemplos de aplicaciones

Una de las primeras aplicaciones de los mapas autoorganizados fue propuesta por Kohonen en la que se describe una *máquina de escribir fonética (neural phonetic typewriter)* [33] cuya habilidad es interpretar y reconocer el habla humana. El aparato utilizado usa un arreglo bidimensional de nodos entrenados con entradas de 15 componentes que representan el análisis espectral de palabras habladas, las cuales son tomadas cada 9.83 milisegundos. Para generar estos vectores de entrada se necesita un preprocesamiento realizado al sonido en el que se utilizan un micrófono que reduzca el ruido del ambiente, convertidor analógico-digital de 12 bits, transformada rápida de Fourier de 256 puntos llevada a cabo cada 9.83 milisegundos (la que permite agrupar el canal espectral en 15 grupos), así como otros filtros y normalización realizados al vector resultante por la trasformada. Las neuronas dentro de la malla bidimensional son agrupados durante la etapa de entrenamiento y el mapa resultante (Figura 2.13) es calibrado mediante el espectro de los fonemas (vectores de entrada). Cada vez que una palabra es dicha, se realiza el análisis de ella y se envía al SOM como una secuencia de vectores para obtener las neuronas excitadas (correspondientes a un fonema) permitiendo trazar un camino correspondiente a la secuencia de patrones de entrada. El camino obtenido es de hecho, la transcripción de los fonemas (Figura 2.14).

Otros problemas en los que SOM ha dejado huella han sido en el monitoreo de las condiciones tanto de plantas como procesos, clasificación de nubes, análisis de datos de micro arreglos de DNA y clasificación y organización de imágenes [62].

La agrupación de características (*clustering*) ha sido beneficiada al explotar el mapa topológico obtenido por el SOM; lo que se desea obtener es subconjuntos con datos similares. Un ejemplo de esto es la organización y clasificación de documentos. S. Kaski et al. [28, 36] proponen un método de organización de documentos utilizando el algoritmo SOM, llamado *WEBSOM* en el que cada documento se representa por un vector de ocurrencias de palabras clave (cada

documento lleva un preprocesamiento previo) y los documentos son agrupados de acuerdo a sus similitudes.

2.7. Resumen

Los mapas de Kohonen son una forma de aprendizaje no supervisado competitivo en donde cada neurona de la malla compite entre sí para determinar cuál se va a actualizar. Esta competencia define cómo se van a agrupar los datos de entrada sin necesidad de "maestro" que supervise cómo debe realizarse.

El trabajo de Kohonen está inspirado en el cerebro que realiza un mapeo de las señales obtenidas por los diferentes órganos sensoriales a la corteza cerebral, en donde diferentes áreas de la corteza controlan las diferentes actividades sensoriales.

La idea básica es del algoritmo de aprendizaje es:

- Presentar la entrada.
- Encontrar la unidad ganadora.

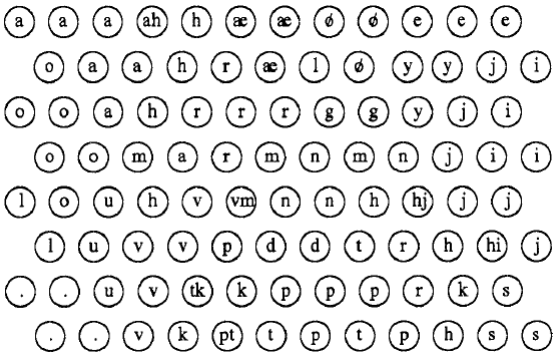


Figura 2.13: El mapa fonético en donde se muestran las neuronas (círculos) etiquetadas con los símbolos de los fonemas con los que han respondido a una mejor respuesta. (Imagen tomada de *The Self-Organizing Map* [34]).

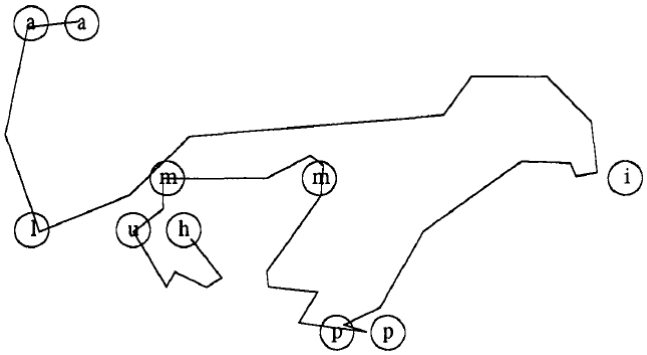


Figura 2.14: Camino que muestra la respuesta a la palabra finlandesa *humpilla*. (Imagen tomada de *The Ñeural'Phonetic Typewriter* [33].)

- actualizar los pesos de la unidad así como la de sus vecinos dentro de la vecindad.
- Repetir los pasos anteriores hasta llegar al máximo número de épocas.

La actualización se realiza mediante el producto de la tasa de aprendizaje y la diferencia entre los vectores de entrada y de pesos sinápticos. La tasa de aprendizaje es un parámetro que permite el control de correcciones que efectuará el aprendizaje. También se tiene el parámetro de vecindad que permite controlar la influencia que tiene cada neurona ganadora con sus vecinas. Ambos parámetros decaen con el tiempo. En las primeras etapas permiten que las correcciones realizadas sean grandes y que conforme avanza el tiempo de entrenamiento, se vayan haciendo graduales para obtener una convergencia.

El trabajo de Kohonen ha sido de utilidad en diferentes áreas y de especial utilidad para el *clustering*. Así mismo, se han derivado diferentes algoritmos que son mejoras al original propuesto por Kohonen, o bien, lo utilizan como base.

Capítulo 3

Reducción de variables

Dentro de un espacio muestral, una variable, llamada también característica, rasgo o atributo, representa una propiedad de un sistema el cual ha sido observado y medido [63]. La reducción de estas variables permite que se tenga un espacio muestral semejante al original, con la diferencia que el nuevo espacio esté representado por un número de variables menor y que contenga la mayor información posible de los datos [26], ya que no todas las variables son necesarias para precisar la clasificación. De esta manera, un espacio de m dimensiones es reducido a uno de $p < m$.

Para la reducción de variables se puede utilizar *extracción de características* o *selección de características* y aunque ambas son muy similares y sus objetivos son reducir la dimensión del espacio muestra, se puede encontrar la diferencia entre ambas.

Mediante la extracción de características se pretende encontrar nuevas características mapeando el conjunto original de características a uno nuevo mediante funciones lineales o no lineales [30, 64, 66, 52, 69, 15] y algunos de los métodos comúnmente utilizados de este tipo son *Principal Component Analysis (PCA)* (Sección 3.2), *Linear Discriminant Analysis (LDA)* y *Singular Value Decomposition (SVD).*

Por otra parte, la selección de características es un proceso que permite identificar el subconjunto más efectivo a ser utilizado [30, 64, 66, 52, 69, 15] para la resolución de un problema y que éste mantenga la mayor precisión de clasificación [52]. Mediante los métodos que involucran este tipo de técnica, el subconjunto seleccionado no contiene ninguna alteración de los datos debido a algún mapeo, sino que únicamente eligen mediante alguna función de discriminación las variables [66].

La selección de características se puede dividir en *filtros*, *envolventes (wrappers)* o *incrustados (embedded).* Los métodos basados en filtros utilizan medidas estadísticas [70, 64, 24] que permitan evaluar qué tan óptima es la característica. Cada característica es evaluada individualmente en base a su relevancia y los mejores evaluados son aquellos que se eligen [70, 66] lo que permite que sean inde-

pendientes al modelo de aprendizaje. Algunos métodos son la prueba χ^2, métodos basados en teoría de la información, selección basada en correlación, entre otros.

Los métodos envolventes realizan una búsqueda a través del espacio de las posibles combinaciones [66] y las métricas de desempeño de cada subconjunto (dentro del espacio de posibles combinaciones) son realizadas por medio de algoritmos predictivos o algoritmos de entrenamiento específicos [66, 24, 64, 70]. Los seleccionados son aquellos que contienen las evaluaciones óptimas. Algunos métodos son SFS, SBS, Plus-l-Minus-r, algorimtos genéticos, entre otros.

Los métodos incrustados mantienen la misma idea que los envolventes, sin embargo, realizan la selección durante la etapa de entrenamiento del clasificador utilizado [64, 66]. Las características son calificadas de acuerdo a su eficacia y penalizadas conforme más grande sea el subconjunto de características seleccionadas guiando al algoritmo a encontrar los subconjuntos con mejor desempeño (dado por el clasificador) y más compactos [24]. Algunos métodos son árboles de decisión, *Naïve Bayes*, etc.

A comparación de los métodos envolventes o incrustados, los métodos en base de filtros son constitucionalmente más eficaces y simples [24] lo que permiten tratar más fácilmente con conjuntos de datos de dimensiones grandes [66] y además, junto con la facilidad que tiene de tratar con diferentes métodos de aprendizaje, permiten que sean la opción más recurrida para la selección y extracción de características.

La elección de qué variables incluir y cuales descartar no es obvia [25] y el posible número de combinaciones a explorar es factorial, por lo que se necesitan técnicas de selección como algunas mencionadas en las secciones siguientes.

3.1. Criterio de correlación

El criterio de correlación es el método estadístico comúnmente utilizado para la selección de variables. En el se emplea el coeficiente de correlación de Pearson para la elección de variables.

3.1.1. Esperanza, varianza y covarianza

La esperanza de una variable aleatoria X representa el promedio ponderado de todos los valores que una variable aleatoria puede tomar. Matemáticamente se expresa de la siguiente manera:

$$\mathbb{E}[X] = \begin{cases} \int_{-\infty}^{\infty} xf(x)dx & \text{si X es continua} \\ \sum_{x \in X} xp(x) & \text{si X es discreta} \end{cases}$$

donde $f(x)$ y $p(x)$ son la función de densidad y masa probabilista respectivamente.

Para cualquier función $g(x)$, la esperanza se calcula

$$\mathbb{E}[g(X)] = \begin{cases} \int_{-\infty}^{\infty} g(x)f(x)dx & \text{si X es continua} \\ \sum_{x \in X} g(x)p(x) & \text{si X es discreta} \end{cases}$$

La varianza es una medida estadística que indica cómo se esparcen los datos de un conjunto respecto a la media y está definido por

$$s^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n-1}$$

donde X_i es cada dato y $\bar{X}$ es la media de la muestra.

Por otro lado la covarianza mide cómo dos datos dentro de la muestra varían entre ellos respecto a la media y está definido por

$$cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{n-1}$$

La matriz de covarianzas Σ permite observar la dispersión de los datos respecto a la media entre las diferentes variables. Esta matriz es simétrica, cuadrada y la diagonal indica la varianza de la i-ésima entrada puesto que $cov(X,X) = var(X)$. La matriz de covarianzas está definida entonces de la manera siguiente:

$$\Sigma = \begin{pmatrix} var(x_1) & cov(x_1,x_2) & \cdots & cov(x_1,x_m) \\ cov(x_1,x_2) & var(x_2) & \cdots & cov(x_2,x_m) \\ \vdots & \vdots & \ddots & \vdots \\ cov(x_m,x_1) & cov(x_m,x_2) & \cdots & var(x_m) \end{pmatrix}$$

3.1.2. Criterio de correlación

El coeficiente de correlación entre dos variables aleatorias X y Y, denotado por $\rho(X,Y)$, se calcula de la siguiente manera:

$$\rho(X,Y) = \frac{cov(X,Y)}{\sqrt{Var(X)Var(Y}}$$

Este coeficiente indica el grado de linealidad entre X y Y y su valor está en el intervalo $[-1,1]$. Valores negativos indican que Y tiende a incrementar mientras X se incrementa y valores positivos indican que Y se incrementa mientras X también lo hace. Mientras más alejado es el valor de cero, existe mayor linealidad entre X y Y. Cuando $\rho = 0$ no existe ninguna relación entre las variables aleatorias y se dice que no están correlacionadas [55].

Mediante estas correlaciones, aquellos valores cercanos ± 1 son eliminados pues una variable está descrita por la otra.

El criterio de correlación sólo detecta dependencias lineales y por como se calcula, es indispensable que cada una de las características sean variables numéricas.

3.2. Análisis de componentes principales

Mediante el análisis de componentes principales (*PCA*) se busca encontrar combinaciones lineales que permitan maximizar la varianza contenida en el conjunto de datos de entrada [26, 21, 2] y encontrar un subconjunto de menor cardinal al original que lo represente adecuadamente [31].

El conjunto de datos $\mathbf{X}$ consta de las observaciones realizadas a n individuos diferentes. Cada individuo x_i tiene m atributos diferentes y la idea detrás de PCA es encontrar $p \leq m$ atributos que representen a la misma muestra [26, 31].

El objetivo de los componentes principales es minimizar la redundancia dada por la covarianza entre variables y maximizar la varianza de cada variable[57]. El minimizar la covarianza implica encontrar aquellas covarianzas de menor valor, puesto que un valor grande implica redundancia entre ellas.

Para encontrar los componentes principales de $\mathbf{X}$ se asume que $\mathbb{E}[\mathbf{X}] = 0$ [26, 2, 57] por lo que podemos escribir la covarianza de la matriz de datos X como

$$\Sigma = \frac{1}{n-1}\mathbf{X}\mathbf{X}^T \tag{3.1}$$

Lo anterior se debe a que la matriz de covarianzas de dos vectores $\mathbf{a}$ y $\mathbf{b}$ con media cero es posible escribirla como $s^2_{\mathbf{ab}} = \frac{1}{n-1}\mathbf{a}\mathbf{b}^T$, por lo que renombrando $a = x_i$ y $b = x_j$ ($\mathbf{a}$ y $\mathbf{b}$ dos renglones de la matriz de datos $\mathbf{X}$) queda la Ecuación 3.1.

Ahora hay que encontrar una matriz ortonormal P tal que $Y = P\mathbf{X}$ de forma que la matriz de covarianzas de Y sea diagonalizada:

$$\begin{aligned} s^2_Y &= \mathbb{E}[Y^2] && (3.2)\\ &= \mathbb{E}[P\mathbf{X}(P\mathbf{X})^T] && (3.3)\\ &= \mathbb{E}[P(\mathbf{X}\mathbf{X}^T)P^T] && (3.4)\\ &= P\mathbf{X}\mathbf{X}^{\mathbb{T}}P^T && (3.5)\\ &= P\Sigma P^T && (3.6) \end{aligned}$$

Para encontrar la combinación lineal dada por Y con máxima varianza, P debe satisfacer $PP^T = 1$, lo que reduce encontrar la solución al sistema de ecuaciones $\Sigma P = \lambda P$, o bien, encontrar los valores y vectores propios de Σ.

Sean $\lambda_1, \lambda_2, ..., \lambda_m$ los valores propios asociados a Σ y sus vectores propios denotados por $p_1, p_2, ..., p_m$ de forma que $\lambda_1 > \lambda_2 > ... > \lambda_m$. De esta forma, cada fila de P es un vector propio de Σ y tendríamos m ecuaciones de la forma

$$\Sigma p_j = \lambda_j p_j \quad j = 1, 2, ..., m \tag{3.7}$$

y combinándolas nos queda

$$\Sigma P = P\Lambda \tag{3.8}$$

donde Λ es una matriz diagonal dada por los valores propios de Σ. Como cada valor propio de Λ es diferente, podemos escribir $PP^T = I$ por lo que $P^T = P^{-1}$, de esta manera, la Ecuación 3.8 es reescrita como

$$P^T \Sigma P = \Lambda \tag{3.9}$$

Cada vector propio de Σ (o fila de P) representa un componente principal sobre la que se maximiza la varianza y cada *i-ésimo* elemento en la diagonal de la matriz Λ es la varianza de $\mathbf{X}$ sobre p_i.

Como actúa PCA en la reducción de dimensionalidad es eligiendo los primeros $p \leq m$ vectores de P o los primeros p componentes principales. La fila p_1 de P es la dirección en el espacio m-dimensional en la que la varianza de X es maximizada. La fila p_i es otra dirección, perpendicular a p_j $(j < i)$, en la que la varianza es maximizada. La Figura 3.1 muestra lo anterior. Se tienen 100 vectores de dimensión dos (puntos rojos) representando datos ficticios y las líneas azules representan los componentes principales. X_1 es el primer componente principal en el que se destaca la mayor varianza mientras que X_2 es el segundo componente principal.

Este método no es invariante a transformaciones de las variables, por ejemplo, el escalamiento de las variables de entrada [5].

A diferencia de SOM, PCA describe las propiedades estadísticas globales de la distribución de datos dadas por los componentes principales, mientras que SOM tiene dos direcciones principales encontradas considerando las diferencias entre los vectores vecinos [35]. Más direcciones pueden obtenerse agregando más dimensiones a la red.

3.3. Información mutua

Para poder tratar con la información mutua es necesario hablar primero de la entropía y su significado. La entropía fue introducida por Claude Shanon en 1948 en la que expuso los fundamentos de la teoría de la información. Esta teoría permite el estudio de la eficiencia de la representación de la información y sus limitaciones de transmisión sobre un canal de comunicación [26] cuyo impacto directo fue dirigido al diseño de sistemas de comunicación que fueran eficientes y fiables y extendido a diferentes aplicaciones como la selección de variables, la cual se trata en esta sección.

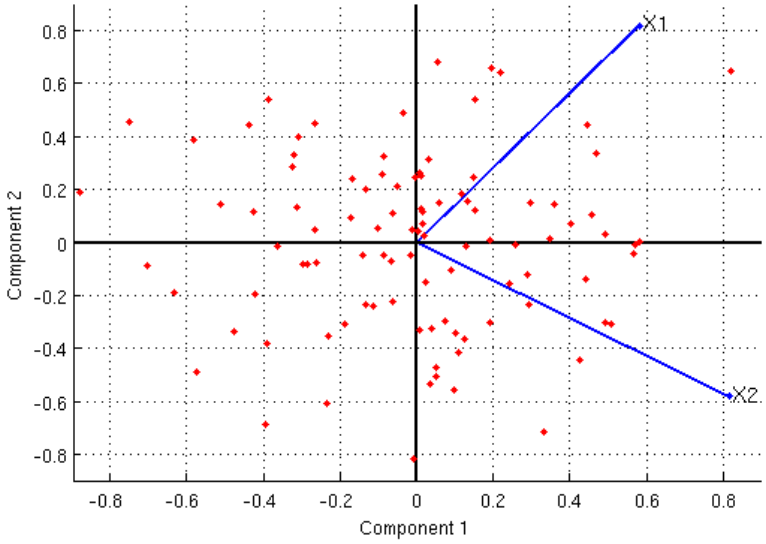


Figura 3.1: PCA. La figura muestra los dos componentes principales (líneas azules) X_1 y X_2 de una distribución aleatoria normal con media 0 y desviación estándar 1. X_1 es el primer componente principal cuya varianza es máxima. X_2 es el segundo componente principal con varianza máxima y ortonormal al primer componente.

3.3.1. Entropía, entropía conjunta y entropía condicional

La entropía es una medida de incertidumbre de una variable aleatoria [13, 63] la cual está relacionada con la probabilidad de ocurrencia de un evento [63]. Con esta medida, podemos conocer la cantidad esperada de información contenida en una variable aleatoria [26].

La cantidad de información adquirida después de observar un evento está definido como:

$$I(x) = \log\left(\frac{1}{p(x)}\right) = -log(p(x)) \tag{3.10}$$

donde $p(x)$ es la función de masa probabilista de la variable aleatoria discreta X $(p(x) = \mathbb{P}[X = x])$.

La base del logaritmo es arbitraria. Cuando se utiliza el logaritmo natural, las unidades de información son *nats* mientras que para base dos, que es la más común, son *bits* [26].

Las propiedades para la ecuación 3.10 son las siguientes y no dependen de la base del logaritmo [26]:

- $I(x) = 0$ si $p(x) = 1$ por lo que no se tiene ninguna adquisición de información ya que el resultado siempre es conocido y no existen "sorpresas" [26].
- $I(x) \geq 0$ por lo que siempre tendremos alguna o ninguna ganancia de información, pero nunca pérdida de la misma.

- Menor probabilidad de ocurrencia de un evento implica mayor información ganada: $p(x_i) < p(x_j) \Rightarrow I(x_i) > I(x_j)$

Teniendo expresada la ganancia de información se pude expresar la entropía de una variable aleatoria como la esperanza de la ganancia de la información, es decir:

$$H(X) = \mathbb{E}[I(X)] \tag{3.11}$$
$$= \sum_{x \in X} p(x) I(x) \tag{3.12}$$
$$= -\sum_{x \in X} p(x) \log p(x) \tag{3.13}$$

Si la variable aleatoria es continua con función de densidad $f_X(x)$, la entropía es expresada como:

$$H(X) = \int_{-\infty}^{\infty} f_X(x) \log f_X(x) dx \tag{3.14}$$
$$= -\mathbb{E}[\log f_X(x)] \tag{3.15}$$

La entropía conjunta para dos variables aleatorias discretas X, Y con función de masa conjunta $p(x, y)$ está definida como

$$H(X, Y) = -\mathbb{E}[\log p(x, y)] \tag{3.16}$$
$$= -\sum_{x \in X} \sum_{y \in Y} p(x, y) \log p(x, y) \tag{3.17}$$

que es la suma de incertidumbre contenida por ambas variables y su valor está en el rango $\text{máx}(H(X), H(Y)) \leq H(X, Y) \leq H(X) + H(Y)$ [63].

El valor máximo ocurre cuando las variables aleatorias X y Y son independientes mientras que el valor mínimo ocurre cuando X es dependiente de Y.

La entropía condicional de dos variables aleatorias discretas X y Y representa la cantidad de incertidumbre del sistema X después de haber observado al sistema Y [26] y está dado por:

$$H(X|Y) = \mathbb{E}[-\log \mathbb{P}[X|Y]] \tag{3.18}$$
$$= -\sum_{y\in Y} p(y)\mathbb{E}[\log \mathbb{P}[X|Y=y]] \tag{3.19}$$
$$= -\sum_{y\in Y} p(y) \sum_{x\in X} p(x|y)\log p(x|y) \tag{3.20}$$
$$= -\sum_{y\in Y}\sum_{x\in X} p(y)p(x|y)\log p(x|y) \tag{3.21}$$
$$= -\sum_{y\in Y}\sum_{x\in X} p(x,y)\log p(x|y) \tag{3.22}$$

Otra forma de obtener la entropía condicional es [26, 13]:

$$H(X|Y) = H(X,Y) - H(Y) \tag{3.23}$$

La ecuación 3.23 se obtiene del teorema de la regla de la cadena [13] desarrollado en lo siguiente:

$$H(X,Y) = -\mathbb{E}[\log p(x,y)] \tag{3.24}$$
$$= -\sum_{x\in X}\sum_{y\in Y} p(x,y)\log p(x,y) \tag{3.25}$$
$$= -\sum_{x\in X}\sum_{y\in Y} p(x,y)\log(p(x)p(y|x)) \tag{3.26}$$
$$= -\sum_{x\in X}\sum_{y\in Y} p(x,y)\log p(x) - \sum_{x\in X}\sum_{y\in Y} p(x,y)\log p(y|x) \tag{3.27}$$
$$= -\sum_{x\in X} p(x)\log p(x) - \sum_{x\in X}\sum_{y\in Y} p(x,y)\log p(y|x) \tag{3.28}$$
$$= H(X) + H(Y|X) \tag{3.29}$$
$$= H(Y) + H(X|Y) \tag{3.30}$$

Ejemplo de entropía condicional

Dada una comunidad se sabe que aproximadamente el 15 % de las familias no tienen hijos, 20 % tiene un hijo, 35 % tienen dos y 30 % tienen tres. Además tener niña o niño es equiprobable y no depende de algún factor. Si se elige al azar una familia, entonces X indica el número de niños y Y el número de niñas. Lo que deseamos obtener es la entropía conjunta de X y Y y la Tabla 3.1 muestra la función de masa probabilista de que la familia tenga niños o niñas.

$p(x)$ y $p(y)$ son las marginales de X y Y respectivamente.

La entropía se obtiene de la siguiente manera:

x \ y	0	1	2	3	$p(x)$
0	0.15	0.1	0.0875	0.375	0.375
1	0.1	0.175	0.1125	0	0.3875
2	0.0875	0.1125	0	0	0.2
3	0.0375	0	0	0	0.0375
$p(y)$	0.375	0.3875	0.2	0.0375	

Tabla 3.1: Distribución conjunta de X y Y: $\mathbb{P}[X = x, Y = y]$

$$\begin{aligned} H(X) &= -\mathbb{E}[\log p(x)] \\ &= -\left(0.375\log(0.375) + 0.3875\log(0.3875) + 0.2\log(0.2) + 0.0375\log(0.0375)\right) \\ &= 1.7027 \end{aligned}$$

Análogamente

$$\begin{aligned} H(Y) &= -\mathbb{E}[\log p(y)] \\ &= 1.7027 \end{aligned}$$

Para la entropía conjunta:

$$\begin{aligned} H(X,Y) &= -\sum_{x\in X}\sum_{y\in Y} p(x,y)\log p(x,y) \\ &= -(0.15\log(0.15) + 0.1\log(0.1) + 0.0875\log(0.0875) + 0.375\log(0.375) \\ &+ 0.1\log(0.1) + 0.175\log(0.175) + 0.0875\log(0.0875) \\ &+ 0.1125\log(0.1125) + 0.0375\log(0.0375)) \\ &= 3.1929 \end{aligned}$$

Para las entropías condicionales:

$$\begin{aligned} H(Y|X) &= H(X,Y) - H(X) \\ &= 3.1929 - 1.7027 \\ &= 1.4902 \end{aligned}$$

$$\begin{aligned} H(X|Y) &= H(X,Y) - H(Y) \\ &= 3.1929 - 1.7027 \\ &= 1.4902 \end{aligned}$$

En este caso, $H(X|Y) = H(Y|X)$ ya que $(X) = H(Y)$.

3.3.2. Información mutua

Mediante la información mutua *(MI)* es posible explicar cómo se reduce la incertidumbre de una variable aleatoria debido al conocimiento de otra [13] por lo que permite conocer la dependencia existente entre ambas variables[13, 12]. Permite observar cualquier tipo de relaciones, incluyendo las no lineales, entre variables aleatorias [63, 13] y es invariante a las transformaciones del espacio de características como rotación, traslación, etcétera [63].

La información mutua se calcula de la siguiente manera:

$$
\begin{aligned}
I(X;Y) &= H(X) - H(X|Y) && (3.31)\\
&= -\sum_{x\in X} p(x)\log p(x) + \sum_{x\in X}\sum_{y\in Y} p(x,y)\log p(x|y) && (3.32)\\
&= -\sum_{x\in X}\sum_{y\in Y} p(x,y)\log p(x) + \sum_{x\in X}\sum_{y\in Y} p(x,y)\log p(x|y) && (3.33)\\
&= \sum_{x\in X}\sum_{y\in Y} p(x,y)\log\left(\frac{p(x|y)}{p(x)}\right) && (3.34)\\
&= \sum_{x\in X}\sum_{y\in Y} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right) && (3.35)
\end{aligned}
$$

En el caso de que las variables aleatorias sean continuas, la información mutua se calcula de manera semejante que en la ecuación 3.31:

$$
I(X;Y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{X,Y}(x,y)\log\left(\frac{f_{X,Y}(x,y)}{f_X(x)f_Y(y)}\right)dxdy \qquad (3.36)
$$

La información mutua entre dos variables aleatorias es simétrica, por lo que $I(X;Y) = I(Y;X)$. Es positiva y sólo es cero cuando X y Y son independientes y mantiene las siguientes relaciones [26, 63, 39]:

$$
\begin{aligned}
I(X;Y) &= H(X) - H(X|Y) && (3.37)\\
I(X;Y) &= H(Y) - H(Y|X) && (3.38)\\
I(X;Y) &= H(X) + H(Y) - H(X,Y) && (3.39)\\
I(X;Y) &= I(Y;X) && (3.40)\\
I(X;X) &= H(X) && (3.41)
\end{aligned}
$$

La figura 3.2 muestra mediante diagrama de Venn cómo se relacionan las entropías y la información mutua entre variables aleatorias. El círculo azul muestra la entropía de X, mientras que el gris la de Y. La intersección de ambos es la información mutua y la unión muestra la entropía conjunta.

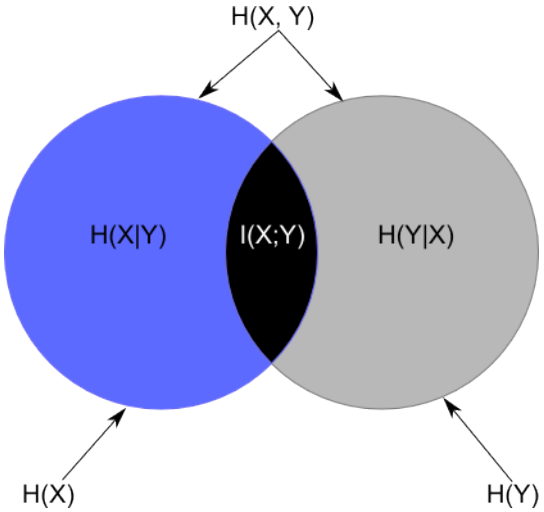


Figura 3.2: Diagrama de Venn que muestra la relación entre la entropía y la información mutua. El círculo del lado izquierdo (contorno azul) muestra la entropía de X. El círculo derecho (con contorno gris), la entropía de Y. El sombreado azul del círculo izquierdo muestra la entropía de X dado el conocimiento de Y y el sombreado gris del círculo derecho la entropía de Y dado el conocimiento de X. La unión de los dos círculos es la entropía conjunta de X y Y. La parte central de la figura, el sombreado negro, muestra la información mutua de X con Y.

Ejemplo

Continuando con el Ejemplo en la sección 3.3.1, podemos calcular la información mutua contenida por las variables X y Y mediante la ecuación 3.31:

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= 1.7027 - 1.4902 \\ &= 0.2125 \end{aligned}$$

3.3.3. Selección de variables

La información mutua se guía por la dependencia estadística entre dos variables aleatorias por lo que es posible utilizarla para la selección características [4].

Existen diferentes métodos de selección de variables utilizando información mutua y las áreas en las que se han utilizado son diversas como en la selección de datos de entrada para las redes neuronales [5, 38], selección de señales médicas, visualización de datos, entre otros [4].

Battiti [5] propone el método *Mutual Information Based Feature Selection (MIFS)* para la selección de variables en el que calcula la información mutua entre características individuales y la clase a la que pertenece mediante una búsqueda *greedy* permitiendo eliminar el análisis de todos los posibles subconjuntos de características. La idea básica es elegir (una a la vez) la característica que maximice la información respecto a la clase y sustrayendo una cantidad proporcional a la información mutua promedio de aquellas ya seleccionadas.

Battiti describe el algoritmo MIFS [5] de la siguiente manera:

Algoritmo 3.1: Algoritmo MIFS

1. $F \leftarrow \{f_1, f_2, \cdots, f_n\}$ Conjunto de características iniciales.
   $S \leftarrow \{\}$. Conjunto de características elegidas.
2. Calcular la información mutua con la clase de salida:
   $\forall f \in F$ calcular $I(C; f)$
3. Encontrar la característica f que maximiza $I(C; f)$
   $F \leftarrow F \setminus \{f\}$
   $S \leftarrow \{f\}$
4. Selección greedy. Reperir hasta que $|S| = k$:
   a. $\forall (f, s), f \in F, s \in S$ calcular $I(f, s)$
   b. Selección de la siguiente característica:
      Elegir la característica f que maximiza
      $G = I(C; f) - \beta \sum_{s \in S} I(f; s)$
      $F \leftarrow F \setminus \{f\}$
      $S \leftarrow S \bigcup \{f\}$

En el algoritmo 3.1, cada $f \in F$ es un vector aleatorio, como X en la ecuación 3.31, que describe una característica de la muestra, por lo que F es un conjunto de vectores aleatorios; C representa la variable aleatoria que contiene la información sobre las clases de los datos de entrada. El parámetro β (parámetro de redundancia) regula la importancia de la información mutua entre la característica candidata y las seleccionadas; si es cero, sólo se considera la información mutua con la clase de salida. De acuerdo con Battiti, el valor apropiado de este parámetro se encuentra entre 0.5 y 1.

Al tener dos variables f y s altamente dependientes entre sí, la información mutua $I(f; s)$ es muy alta y después de haber elegido a la mejor, la selección de la otra es penalizada [5]. De esta forma, la variable a elegir debe brindar la suficiente información de la clase a la que pertenece y mantenerse separada de las variables ya elegidas.

A partir del trabajo de Battiti, diferentes autores han seguido bajo la misma línea y en varios casos los métodos resultantes son modificaciones al trabajo original de Battiti. Kwak y Choi [39] realizan una modificación al método *MIFS*, en el que modifican la selección greedy. El nuevo método es *MIFS-U* y el cambio realizado es la ecuación de selección del paso 4b por la ecuación 3.42.

$$G = I(C; f) - \beta \sum_{s \in S} \frac{I(C; s)}{H(s)} I(f; s) \tag{3.42}$$

Otras modificaciones son los algoritmos *mRMR* propuesto por Peng, Long y Ding [49], *NMIFS* por Estévez, Tesmer, Pérez y Zurada [15] y *NIMFS-FS2* por Cang y Yu [11], en los que al igual que *MIFS-U*, cambian la selección greedy de 3.1 por las ecuaciones 3.43 y 3.44 respectivamente.

$$G = I(C; f) - \frac{1}{|S|} \sum_{s \in S} I(f, s) \tag{3.43}$$

$$G = I(C; f) - \frac{1}{|S|} \sum_{s \in S} \left(\frac{I(f, s)}{\text{mín}\{H(f), H(s)\}} \right) \tag{3.44}$$

En 3.43 se pretende evaluar tanto la redundancia como la relevancia de una característica mediante los criterios de *Min-Redundancia* y *Max-Relevancia* respectivamente[1]. El criterio de máxima relevancia busca aquellas características que maximizan la relevancia de un subconjunto S con respecto a la clase c y está dada por

$$\max D(S,c), D = \frac{1}{|S|}\sum_{x_i \in S} I(x_i, c) \tag{3.45}$$

El criterio de mínima redundancia requiere la eliminación de características dependientes entre sí, por lo que sólo elige características mutuamente excluyentes mediante

$$\min R(S), R = \frac{1}{|S|^2}\sum_{x_i, x_j \in S} I(x_i, x_j) \tag{3.46}$$

Al combinar los dos criterios previos mediante la función $\Phi = D - R$ se obtiene la característica que optimiza simultáneamente los criterios de máxima relevancia y mínima redundancia, es decir, $\max \Phi$ permite obtener la ecuación 3.43.

La ventaja de *mRMR*, *NMIFS* y *NMIFS-FS2* sobre *MIFS* y *MIFS-U*, es que no requieren del parámetro de redundancia β, el necesita de diferentes ejecuciones de los algoritmos para encontrar un valor adecuado.

Otros métodos basados en información mutua se han propuesto como los descritos en [70, 4, 73].

3.3.4. Estimación de información mutua

Cuando las variables aleatorias de las que se requiere obtener la MI son discretas, las funciones de masa probabilistas son obtenidas mediante conteos e histogramas. Por ejemplo, Si X puede tener tres valores diferentes en una muestra de n valores, entonces la probabilidad de cada valor es el cociente entre el número de apariciones de éste y el tamaño de la muestra. De esa manera se pueden conocer las probabilidades. Sin embargo, cuando al menos una de las dos variables aleatorias involucradas para calcular MI es continua, el cálculo se complica, por lo que es necesario introducir algún mecanismo de hacer discretos de los valores continuos mediante, por ejemplo, estimación de la densidad [49], puesto que, generalmente, se desconoce la función de densidad. Una manera de lograr esta estimación es mediante ventanas de Parzen (*Parzen Windows*) [23, 38, 49].

Mediante la ventana de Parzen se puede estimar la función de densidad de un vector aleatorio continuo. Dadas n muestras de un vector d dimensional x, la función de densidad aproximada de $p(x)$ está por

[1]Los términos en inglés son*Min-Redundancy* y *Max-Relevance* respectivamente, que conforman al método *mRMR* de Peng [49].

$$\hat{p}(x) = \frac{1}{n}\sum_{i=1}^{n}\phi(x - x_i, h) \tag{3.47}$$

donde x_i es la $i-ésima$ muestra, ϕ es la "función ventana" y h es un parámetro que define el ancho de la ventana. Al elegir correctamente ϕ y h, $\hat{p}(x)$ converge a la densidad real cuando n tiende a infinito [38, 49].

Comúnmente se utiliza la función gaussiana como función ventana, dada por

$$\phi(z, h) = \frac{1}{(2\pi)^{d/2}h^d|\Sigma|^{1/2}}exp\left(-\frac{z^T\Sigma^{-1}z}{2h^2}\right) \tag{3.48}$$

donde $z = x - x_i$, d es la dimensión de la muestra x y Σ es la matriz de covarianzas de z. Cuando $d = 1$, la ecuación 3.47 es la función marginal y cuando $d \geq 2$ se tiene una función de densidad multivariada.

3.4. Mapas autoorganizados

La selección de variables propuesta por Benabdeslem [7] (*HI-SOM*) hace uso de mapas autoorganizados para encontrar el subconjunto óptimo de variables. Para realizar esta selección es necesario entrenar los datos $d - 1$ veces, donde d es el número de variables del conjunto de datos original. Cada entrenamiento de datos se realiza con una variable menos que en el paso anterior, por lo que se requiere de una métrica para decidir cuál variable se debe eliminar (la que menos información brinda al sistema o que menor importancia tiene).

Benabdeslem indica la importancia de una variable $j \in \{1, 2, ..., d\}$ asociada a cada neurona c mediante

$$h_{jc} = \gamma_c \frac{w_{cj}}{\sum_{k=1}^{d} w_{ck}} \tag{3.49}$$

donde $\gamma_c = \frac{frecs(c)}{N}$, $frecs(c)$ es el número de elementos asignados a la neurona c, N el tamaño de los datos de entrenamiento y w_{ck} es la entrada k del vector de pesos de la neurona c.

La importancia de una variable j asociada a todo el conjunto de neuronas está definida por

$$\bar{h_j} = \sum_{c \in C} h_{jc} \tag{3.50}$$

De esta manera se evalúa la distribución de las muestras en las neuronas y de cada característica con el resto. Si una neurona no fue estimulada por algún dato

de entrada, entonces no existe influencia de variable alguna, por lo que γ_c es un parámetro que adecua la proporción de influencia de la variable d con el número de entradas a las que responde la neurona c.

El algoritmo de selección consiste en dos pasos.

El primero es la etapa de aprendizaje de *HI-SOM* y consiste en los siguientes pasos:

Algoritmo 3.2: Algoritmo de selección mediante mapas autoorganizados (*HI-SOM*)

1. p=1, $S_p = \{1, 2, ..., d\}$ (todas las variables)
2. Crear un mapa C_p mediante SOM
   con las variables de S_p
3. Calcular $\bar{h_j}$, $1 \leq j \leq d$
4. Eliminar la variable j asociada al menor $\bar{h_j}$
   $S_{p+1} = S_p \setminus \{j\}$
5. Evaluar la calidad del mapa.
6. Repetir desde (2) hasta la última variable

La segunda etapa consiste en elegir el conjunto de variables. El algoritmo 3.2 genera $d - 1$ mapas, cada uno con una variable menos. La selección de las variables corresponde a determinar qué mapa tiene mejor comportamiento eligiendo aquel cuyo error topológico es menor, es decir, el que mantiene un mayor orden topológico. Si dos mapas son los que mantienen el mismo error, se elige el que tiene el mayor "error global decreciente (GDR)" definido como [7]:

$$GDR = \frac{E_i - E_f}{E_i} \tag{3.51}$$

donde E_i y E_f son los errores de cuantificación inicial y final respectivamente que permiten medir la distancia entre un vector de entrada y la neurona ganadora antes y después del entrenamiento. De esta manera, GDR obtiene una proporción de acercamiento del vector de entrada a su neurona ganadora después del entrenamiento.

Este error sirve para discriminar, entre dos mapas de Kohonen con igual preservación topológica, aquel que mejor representa a los datos.

SOM permite agrupar los datos de acuerdo a similitudes entre ellos. Sin embargo, al eliminar variables es posible que el orden que preserva aumente debido a que la variable eliminada ocasiona ruido al sistema, o bien, que el orden disminuya debido a que la variable al ser eliminada, pese a tener menor importancia, rompa el equilibrio existente. Esta es la razón por la que se deben generar tantos mapas de Kohonen y calcular sus errores.

3.5. Sequential Search

La selección de características mediante este tipo de técnicas consiste en ir generando el subconjunto de características creciente o decrecientemente. Usualmente son técnicas *greedy* por lo que no garantizan encontrar un resultado óptimo

global de las características seleccionadas, sino solamente locales y su complejidad suele ser de $O(n^2)$ en el peor caso [14].

Al iniciar la búsqueda con un conjunto vacío e ir añadiendo características, que maximicen el criterio de evaluación, iterativamente una a la vez hasta tener la dimensión deseada, se tiene el método *Sequential Forward Selection (SFS)* [68]. Si el conjunto inicial es el conjunto completo de características disponibles y se eliminan iterativamente, análogamente a *SFS*, se obtiene el método *Sequential Backward Selection* [41].

La desventaja de estos métodos es el "efecto de anidamiento" en el que las características descartadas (en el caso de SBS) no pueden volver a ser elegidas, mientras que las características elegidas (en el caso de SFS) no pueden ser despreciadas en pasos posteriores [50]. El método *Plus-l-Minus-r (LRS)* desarrollado por [58] permite eliminar este efecto mediante dos valores de control l y r. Si $l > r$, LRS inicia con un conjunto vacío y repetidamente agrega l características y elimina r. Si por el contrario, $l < r$, LRS inicia con el conjunto de características totales y repetidamente elimina r características y agregar l. Aunque permite eliminar el efecto de anidamiento de SBS y SFS, no existe alguna manera capaz de encontrar valores óptimos para l y r [40].

3.5.1. Sequential Floating Search

Pudil y Novovičová ([50]), describen dos tipos de búsqueda flotante secuencial: *Sequential Forward Floating Selection* y *Sequential Backward Floating Selection* cuya idea parte de las búsquedas secuenciales SFS y SBS.

Aquí se describen dos conjuntos: el conjunto $X_k = \{x_i | 1 \leq i \leq k, x_i \in Y\}$ que es el conjunto de k características seleccionadas del conjunto $Y = \{y_i | 1 \leq i \leq D\}$ de D características disponibles.

La característica más significante f^+ del conjunto X_k se tiene cuando:

$$f^+ = \underset{f \in Y \setminus X_k}{\text{máx}} J^+(X_k, f)$$

es decir, f^+ es aquella característica que maximiza la función de evaluación J al agregar $f \in Y \setminus X_k$ en X_k, obteniendo así el nuevo conjunto X_{k+1}.

Análogamente, la característica menos significante f^- del conjunto X_k se obtiene cuando:

$$f^- = \underset{f \in X_k}{\text{máx}} J^-(X_d, f)$$

es decir, f^- es aquella característica que maximiza la función de evaluación J al eliminar $f \in X_k$ de X_k, obteniendo así el nuevo conjunto X_{k-1}.

Sequential Forward Floating Selection (SFFS)

El algoritmo SFFS va descubriendo nuevas características al aplicar el procedimiento básico de SFS y eliminaciones de la peor característica en el nuevo subconjunto. La idea básica es partir del método SFS para ir agregando características significantes x_{x+1} al conjunto X_k y de ahí, eliminar las menos significantes del nuevo conjunto X_{k+1}. El algoritmo 3.3 describe los pasos.

Algoritmo 3.3: Algoritmo SFSS

1. $k = 0$
   $X_k = \emptyset$
2. (Inclusión) Agregar mediante SFS x_{k+1}, obteniendo:
   $X_{k+1} = X_k + x_{k+1}$
3. (Exclusión condicional) Encontrar la característica menos significante x_s en X_{k+1}.
   Si $J(X_{k+1} - x_s) \geq J(X_{k+1} - x_j)\ \forall j = 1, ..., k,\ x_s = x_{k+1}$:
     $k = k + 1$ y regresar a 2.
   Si $J(X_{k+1} - x_s) > J(X_k)$ para $1 \leq s \leq k$:
     $X'_k = X_{k+1} - x_s$
     Si $k = 2$:
       X_k = X'_k, regresar a 2
4. (Continuación de la exclusión condicional) Encontrar la característica menos significante x_s en X'_k.
   Si $J(X'_k - x_s) \leq J(X_{k-1})$:
     $X_k = X'_k$, regresar a 2
   Si $J(X'_k - x_s) > J(X_{k-1})$:
     $X'_{k-1} = X'_k - x_s$
     $k = k - 1$
     Si $k = 2$:
       $X_k = X'_k$, regresar a 2.
     en otro caso:
       repetir 4.

Sequential Backward Floating Selection (SBFS)

Este método es análogo a SFFS en el sentido que primero elimina la característica menos significante mediante SBS para después, de aquellas excluidas, agregar las más significantes con respecto al nuevo conjunto obtenido por la exclusión.

3.6. Otros métodos

Los algoritmos genéticos también son utilizados para la selección. Se genera una población de individuos, cada uno con una combinación de características, y se evalúan aplicando una función fitness capaz de encontrar aquellas características que mejor describan al conjunto de datos [40, 52, 59].

Otro método de uso común es mediante la prueba de Ξ^2 que permite estudiar el nivel de dependencia entre dos eventos. Para la selección de variables, los dos eventos a probar son la ocurrencia del término y la ocurrencia de la clase a la que pertenece [23, 40].

Debido a los límites definidos en esta tesis y a la gran variedad de métodos existentes, éstos no se discuten aquí.

3.7. Resumen

En este capítulo se ha visto una breve revisión de métodos para reducir el número de variables de una muestra.

La reducción de variables puede ser realizada mediante extracción de características, en donde se aplica una transformación sobre los datos originales para obtener el nuevo subconjunto de datos; y la selección de características en donde se elige un subconjunto de datos que explique mejor el modelo original.

La selección, por su parte, se puede dividir en filtros, en los que se tiene una función, generalmente estadística, capaz de evaluar la relevancia de las características; los envolventes, que realizan búsquedas mediante diferentes combinaciones y sus métricas de desempeño dependen de algoritmos de clasificación específicos; o incrustados, en los que la idea es semejante a la de los envolventes pero la selección es realizada en la etapa de entrenamiento del clasificador utilizado.

PCA es uno de los métodos más populares para la extracción de características y no necesita de una de clases de datos para realizar la extracción. En PCA se busca maximizar la varianza de combinaciones lineales de un conjunto de datos. Este objetivo se logra al encontrar valores y vectores propios, generados por la matriz de correlaciones a partir del conjunto de datos, los cuales permiten encontrar los componentes principales que permiten discriminar variables que pueden ser prescindibles.

La selección de variables mediante información mutua utiliza la dependencia estadística entre dos variables. En diversos métodos utilizados para este fin, se realiza una búsqueda "greedy" en la que la información mutua entre las variables y las clases hacen la discriminación de aquellas a ser seleccionadas: las variables se van seleccionando conforme se maximiza la información mutua entre una característica y una clase de datos, por lo que lo que se busca es maximizar la dependencia entre una clase y una característica.

La información mutua, a diferencia de PCA, permite una descripción adecuada de relaciones no lineales entre los datos. Además, el escalamiento de los datos de entrada no afecta directamente el resultado como sucede en PCA. Por otra parte, al trabajar con información mutua se puede tratar con valores categóricos.

Otros métodos se describen brevemente en este capítulo con el fin de dar un breve panorama sobre la reducción de variables.

Capítulo 4

Caso de Estudio: Codificación de cadenas de DNA

En este capítulo se realiza una breve introducción a la forma y composición química básica de las secuencias de DNA con el fin de poder contextualizar el caso de estudio.

Ya que no es objeto de estudio el entendimiento de las reacciones químicas y moleculares existentes dentro del DNA, este tema así como detalles más especializados son omitidos y pueden ser consultados en la bibliografía utilizada para describir brevemente este capítulo ([67, 29, 65]).

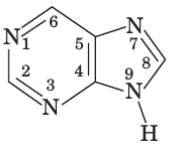
4.1. Estructuras de DNA

En 1869 Fiedrich Miescher descubrió el ácido desoxirribonucleico (DNA), que llamó *nuclei*, aunque su función biológica fue esclarecida años más tarde, en 1944, cuando Oswald Avery, Colin MacLeod y Maclyn McCarty demostraron su importancia: es la molécula encargada de llevar y transmitir el material genético. El poco conocimiento de esta molécula impulsó a James D. Watson y Francis H. C. Crick a proponer en 1953 el modelo de estructura de DNA de doble hélice.

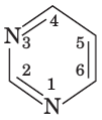
Con el avance en los estudios de esta molécula, ahora se sabe que permite definir un conjunto de instrucciones para que un organismo realice sus diferentes actividades (bilógicamente hablando, la reproducción de sus células, el funcionamiento de cada una, etcétera) y que pueda construir réplicas de sí mismo a nivel celular como de organismo (herencia genética). Además, de acuerdo a Chargraff, la composición básica del DNA de un organismo es la característica del mismo e independiente del que se toma, su edad, o cualquier otra propiedad ambiental [65].

4.1.1. Composición del DNA

La base fundamental del DNA son los nucleótidos, compuestos que permiten construir la cadena de DNA completa. Cada uno de ellos está formado por tres

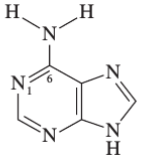


(a) Purina.

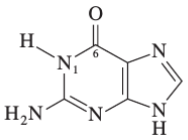


(b) Pirimidina.

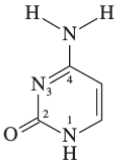
Figura 4.1: Estructuras químicas de la purina y pirimidina.



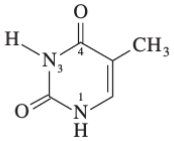
(a) Adenina (A).



(b) Guanina (G).



(c) Citosina (C).



(d) Timina (T).

Figura 4.2: Estructuras químicas las bases nitrogenadas del DNA.

componentes: uno o más grupos de fosfatos, una azúcar conformada por cinco carbonos y una base de nitrógeno que puede ser sustituida por purinas (Figura 4.1a) o pirimidinas (Figura 4.1b). El DNA tiene dos cadenas (o hebras) de estos compuestos llamados polímeros de nucleótidos o polinucleótidos que al ser enrolladas cada una entre sí, se presenta la estructura de doble hélice propuesta por Watson y Crick. Cada una de las hebras pueden ser vistas como un grupo de fosfatos, el azúcar es el medio por el que cada base se adhiere a ella y las uniones entre bases de cada hebra es hecha mediante puentes de hidrógeno.

Las purinas dentro de los ácidos nucleicos son la adenina (A) (Figura 4.2a) y guanina (G) (4.2b) mientras que las pirimidinas son la citosina (C) (Figura 4.2c) y timina (T) (Figura 4.2d). Todos los organismos y células pueden sintetizar los dos tipos de bases ya que son esenciales para el flujo de información genética a transmitir [29].

De acuerdo con el modelo de doble hélice de Watson-Crick y la regla de Chargaff[1], las bases de una hebra de la doble hélice del DNA se liga con bases específicas de la otra hebra: A siempre con T y G siempre con C [29, 65], de esta manera, una purina se enlaza con una pirimidina, lo que mantiene homogénea la anchura entre las dos hebras [45]. Sin embargo la unión de las bases no es homogénea ya que pueden estar ligadas mediante enlaces fuertes, por medio de tres puentes de hidrógeno, como en el caso de C con G o débiles por medio de dos puentes de

[1]La regla de Chargaff dice que el DNA tiene el mismo número de residuos de adenina y timina (A = T) y el mismo número de residuos de guanina y citosina (G = C) [65].

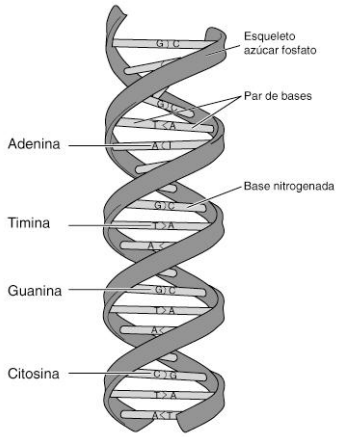


Figura 4.3: Visualización de la estructura del DNA.

hidrógeno como ocurre con T y A. La estructura del DNA se puede apreciar en la Figura 4.3.

En cada una de las bases existen en al menos dos estados alternativos (formas tautoméricas [2]) dependiendo el lado en el que las estructuras de la Figura 4.2 se encuentren. De esta manera la adenina y citosina en su forma convencional se encuentran en su forma amino y en imina en la no convencional, mientras que guanina y timina pueden existir en la forma cetona o enol (Figura 4.4). Además de los dos tautómeros mencionados, se tiene la capacidad para crear alguno alternativo que conlleva a errores durante la síntesis del DNA.

Las bases pueden ser agrupadas en pares de acuerdo a sus características en tres grupos diferentes:

Grupo YR formado por las bases pertenecientes a pirimidinas (Y) que son la citosina y timina; o purinas (R) que son la adenina y guanina.

Grupo WS formado por las bases con enlaces débiles (W) en el caso de timina y adenina; o fuertes (S) caso de guanina y citosina.

Grupo MK formado por el grupo amino (M) que son las bases adenina y citosina; o cetona (K) que ocurre con guanina y timina.

4.2. Codificación

La codificación de las cadenas se realiza mediante "sensores": variables que permiten caraterizar a cada secuencia de DNA [10].

[2]Las palabras tautómero y tautomérico no existen en la lengua española. Sin embargo, serán utilizadas como traducción, aceptada por algunos libros, de las palabras inglesas *toutomer* y *tautomeric* respectivamente.

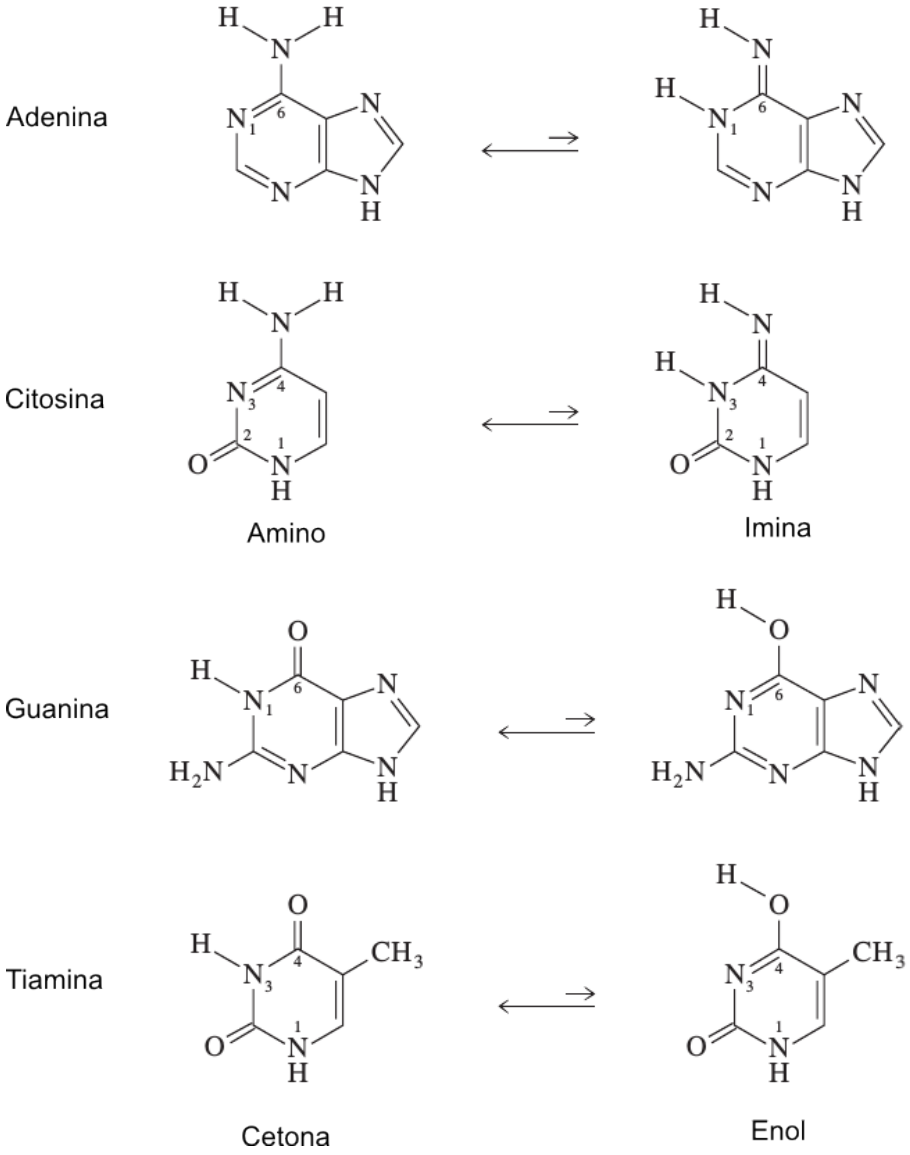


Figura 4.4: Formas tatuméricas de las bases nitrogenadas. Del lado derecho se encuentran las bases en su forma predominante. De este modo, las adenina y citosina pertenecen al grupo de los aminos mientras que la guanina y timina al grupo de las cetonas.

Un tipo de sensor es el IDH (*Index Of DNA Homogeneity*) propuesto por Miramontes *et al.* [44] que pretende expresar la homogeneidad de las secuencias de DNA mediante una traducción binaria de ceros y unos de ella, de esta manera, es posible medir el grado de alternancia de las bases, con el fin de distinguir organismos con diferentes orígenes evolutivos.

Con lo anterior se crean nuevas secuencias codificadas de DNA, una para cada dicotomía: $d(YR)$, $d(WS)$ y $d(MK)$. La codificación se realiza asignando el valor de 0 a las bases Y, W y M y el valor de uno a las bases R, S y K. El índice IDH es calculado a partir de las codificaciones anteriores mediante la ecuación 4.1 para

cada grupo

$$d = \frac{N_{00}N_{11} - N_{01}N_{10}}{N_0 N_1} \tag{4.1}$$

donde $N_{ij},\ i,j = \{0,1\}$ es el número de dinucleótidos ij y N_0 y N_1 son el número de ceros y unos respectivamente. El índice se encuentra dentro del intervalo $[-1,1]$.

En una secuencia donde se tiene aproximadamente la misma cantidad de los cuatros dinucleótidos, es decir, el número de parejas de nucleótidos iguales es semejante al de parejas de nucleótidos diferentes, el índice es cercano a cero. Si dominan las parejas de ceros y unos, el índice es mayor a cero. Si el número de ceros y unos alternados predomina, entonces el índice tiene un valor negativo. Cabe destacar que este valor no es único.

El segundo sensor a utilizar es el contenido $C + G$ que indica la proporción existente de guaninas y citocinas en la cadena y es considerado un elemento que brinda una firma genómica. El contenido permite diferenciar grupos taxonómicos [10]. Por ejemplo, en diferentes especies de bacteria su rango es de 25 %-70 % y en mamíferos de 39 %-46 %. De acuerdo a estudios [72], el alto contenido $C + G$ está altamente correlacionado con las temperaturas en la que un organismo vive y esto es esperado debido a los tres puentes de hidrógeno que une ambas bases. La proporción del contenido $C + G$ es obtenida al sumar el número de bases de citosina y el número de bases de guanina dividido entre el número total de bases de la cadena. La Ecuación 4.2 lo indica, en donde $c(X)$ es el número de bases X y N el total de bases.

$$C + G = \frac{c(C) + c(G)}{N} \tag{4.2}$$

El último sensor está definido por la proporción de cada combinación de pares de bases, que son las 16 posibles combinaciones de dímeros (AA, AC, AG, AT, ..., TA, TC, TG, TT), mediante la Ecuación 4.3

$$xy = \frac{c(XY)}{N(N-1)} \tag{4.3}$$

donde $c(XY)$ es el número de veces que apareció la combinación XY con $X, Y \in \{A, C, G, T\}$ y N es el número total de bases en la secuencia.

4.3. Resumen

El DNA consiste en una estructura de doble hélice de dos cadenas de polinucleótidos. Las cadenas son unidas por puentes de hidrógeno entre las bases de cada cadena siguiendo la regla de Chargaff: C se une con G y A con T. Los enlaces son realizados mediante puentes de hidrógeno y son enlaces fuertes y débiles si los puentes constan de dos o tres átomos de hidrógeno.

La adenina y guanina pertenecen al grupo de purinas (R), mientras que la citosina y timina al grupo de pirimidinas (Y).

Las bases se pueden agrupar de acuerdo a ciertas características. Si se agrupan por el tipo de base (purinas y pirimidinas) es el grupo YR. Si es por el tipo de enlace (fuerte o débil) es el grupo WS. Si se agrupan de acuerdo a las formas tautoméricas (amino o cetona) es el grupo MK.

Las secuencias por sí solas no expresan la información deseada, por lo que es necesario contar con sensores que permitan cuantificar sus propiedades. El primer sensor es el índice de homogeneidad (IDH) con el se puede medir el grado de alternancia de las bases en la secuencia para expresar la homogeneidad de ellas. El IDH es calculado en cada tipo de dicotomía: $d(YR), d(MK), d(WS)$. El segundo sensor a utilizar es el contenido $C + G$ que indica la proporción de guaninas y citosinas en la cadena. EL tercer y último sensor está formado por la proporción de apariciones de un par de bases XY con $X, Y \in \{C, G, T, A\}$. Estos sensores son utilizados en capítulos subsecuentes para mostrar resultados y conclusiones.

Detalles de los procesos que ocurren dentro del DNA así como procesos de replicación, transducción, mutación, etcétera, están fuera del alcance del trabajo.

Capítulo 5

Experimentación y resultados

5.1. Introducción

Esta Sección muestra los experimentos y resultados obtenidos al analizar tres secuencias diferentes de DNA.

5.2. Experimentación

Los experimentos son realizadas en tres etapas:

1. **Preprocesamiento de datos** en donde se leen los archivos que contienen las secuencias de DNA, eliminando caracteres que no pertenezcan a la representación de una base nitrogenada. En este paso se realiza la codificación de los datos mediante los tres tipos de sensores definidos en la Sección 4.2, obteniendo vectores reales con los que se va a trabajar.

2. **Selección de variables** que implica la discriminación de aquellas variables que aporten poca información al sistema de vectores obtenidos en el paso previo.

3. **Clasificación de datos**, paso en el que se buscan las similitudes entre los datos y el agrupamiento de ellos.

5.2.1. Preprocesamiento de datos

La experimentación es realizada mediante la presentación de tres cadenas diferentes de DNA obtenidas de *The National Center for Biotechnology Information (NCBI)*[1]. Las cadenas utilizadas para la experimentación son *Escherichia coli*, *Methanocaldococcus jannaschii* y el cromosoma 21 del *Homo sapiens*.

De cada una de las cadenas son leídos 500 mil bases nitrogenadas. La codificación de las bases leídas se realiza tomando ventanas móviles de mil bases sin traslape entre ellas, como se muestra en la Figura 5.1. Cada una de ellas es

[1]Página web: http://www.ncbi.nlm.nih.gov/

codificada utilizando los tres tipos de sensores diferentes, por lo que al final de este proceso se obtienen 1500 vectores de dimensión 20 (cada cadena de diferente DNA leída es codificada en 500 vectores). Cada vector está formado de la siguiente manera: tres valores corresponden al sensor que mide el IDH, el valor obtenido por el sensor que mide el contenido $C + G$ y el resto son obtenidos por el sensor que mide la proporción de las combinaciones XY con $X, Y \in \{C, G, T, A\}$.

Figura 5.1: Ventanas móviles sin traslape a lo largo de la secuencia de DNA. Los corchetes azules muestran las bases nitrogenadas a ser tomadas en cuenta en el tiempo t. Para el tiempo $t + 1$, la ventana definida por los corchetes azules, se desplaza a la derecha tomando en cuenta las bases que se encuentran entro los corchetes rojos y sin tomar elemento alguno ya utilizado previamente por la ventana anterior.

5.2.2. Selección de variables

Como se ha explicado anteriormente, la selección de variables es un tema muy amplio y estudiado, por lo que existen diversos métodos que pueden ayudar a resolver este problema.

Para este trabajo se realiza la selección mediante el algoritmo propuesto por Benabdeslem en [7], el cual emplea mapas autoorganizados para cumplir con el objetivo. La elección de este método se basa en que es una técnica no supervisada y no se necesita especificar un número de variables a seleccionar, a diferencia de algunos métodos que realizan la selección mediante información mutua. En contraste con PCA, pero de igual manera con aquellos que utilizan información mutua, se mantienen las relaciones no lineales entre los datos.

Una desventaja de este procedimiento es, de hecho, la misma utilización de SOM. Para quedarse con el subconjunto óptimo es necesario entrenar el conjunto de datos $d - 1$ veces, donde d es el número de variables totales, y cada vez se realiza con un SOM de menor dimensión al anterior. Este procedimiento puede ser muy tardado a comparación de otros métodos como los basados en información mutua y su tiempo de ejecución depende de dos parámetros principalmente: el tamaño de la entrada (número de vectores y su dimensión) y la dimensión del mapa neuronal del SOM.

Los parámetros de la selección elegidos para la experimentación en este trabajo son diferentes combinaciones de la dimensión (mallas de 35x35 o 40x40), tasa de aprendizaje (0.9 o 0.1) y número de épocas (700 o 500)[2]. Para fines estadísticos, se realizan 15 simulaciones con cada combinación de parámetros (Tabla 5.1).

La tasa de aprendizaje decrece linealmente con el paso de las épocas hasta un mínimo de 0.001 mediante la ecuación

[2]El radio inicial siempre está dado por el valor de la dimensión.

Combinación	Dimensión SOM	Épocas	Tasa de aprendizaje
C1	35x35	700	0.9
C2	35x35	500	0.9
C3	40x40	700	0.9
C4	40x40	500	0.9
C5	35x35	700	0.1
C6	35x35	500	0.1
C7	40x40	700	0.1
C8	40x40	500	0.1

Tabla 5.1: Combinación de parámetros utilizados durante la discriminación de variables.

$$\alpha(t+1) = \alpha(0) - \frac{t(\alpha(0) - 0.001)}{T} \tag{5.1}$$

donde t es la época actual, $\alpha(0)$ es la tasa de aprendizaje inicial y T es el número de épocas que debe entrenarse el SOM.

Los datos son normalizados[3] dentro del intervalo $[0, 1]$ antes de realizar el procedimiento, ya que, como es indicado por Kohonen [34], esto mejora los resultados numéricos al mantener el mismo rango.

En general, la literatura indica que el valor de la tasa de aprendizaje inicial debe ser grande (cercano a 1) [34, 6], aunque Haykin [26] indica que el valor inicial conviene que sea cercano a 0.1.

El fin de realizar diferentes simulaciones con diferentes parámetros del SOM tiene por objetivo encontrar un subconjunto de variables que representen mejor al conjunto original. Idealmente, los resultados deben ser parecidos entre cada experimentación independientemente de los parámetros utilizados.

5.2.3. Clasificación de datos

Los datos obtenidos por la codificación no dan mucha información por sí solos, o al menos, la información que brindan, no es posible entenderla a simple vista. Los mapas de Kohonen son una herramienta comúnmente utilizada en problemas de este tipo, en el que se desconoce *a priori* la distribución estadística de los datos y donde se requiere encontrar similitudes, posiblemente no lineales entre ellos; son no supervisados en el sentido que no es necesario indicar una salida o clase para que las neuronas sepan cómo reaccionar ante cada estímulo (entrada de datos), por lo que su organización es realizada autónomamente.

[3]El término adecuado debería ser escalamiento de datos, sin embargo, para coincidir con la literatura, se emplea normalización de datos.

Pero para que sea completamente visual en cuanto a la clase que pertenece cada vector, se utiliza un etiquetado de neuronas. Una vez entrenada la red, por cada vector de entrada se encuentra la neurona con mayor excitación a este y a esa neurona se le asigna la clase a la que pertenece el vector de entrada. Una neurona en particular, no necesariamente puede ser mapeada a una sola clase, ya que puede responder a diferentes vectores de entrada y ellos pueden estar asociados a diferentes clases. También es posible que haya neuronas sin etiquetado y esto depende de que haya respondido esa neurona a algún estímulo (dato) o no. Otra manera de realizar este etiquetado con resultados similares es el propuesto en [37], sólo que en este caso cada neurona es asociada al menos a una clase.

Además del etiquetado de neuronas, suele ser de utilidad el modelo de visualización *U-Matrix* [61]. En este modelo se puede ver un mapa neuronal extendido en el que se detallan las distancias existentes entre cada neurona vecina, por lo que es un modelo que permite encontrar *clusters* visualmente. Generalmente este método de visualización pinta cada elemento de la malla de neuronas extendida en escala de grises dando colores más oscuros a mayores distancias y colores más claros a aquellas con distancias más chicas.

Ambos modelos de visualización son utilizados para la clasificación. Ésta se realiza utilizando el subconjunto de datos obtenidos por la selección de variables y el conjunto completo de datos realizando una comparación entre los resultados obtenidos mediante los errores topológicos y cuantitativos de ambas redes entrenadas.

La clasificación se realiza con los siguientes parámetros, en ambos casos:

- Tasa de aprendizaje inicial: 0.9
- Dimensión de la red: 35x35
- Épocas de entrenamiento: 700
- Topología de la red: hexagonal

La tasa de aprendizaje decrece en cada época linealmente mediante la Ecuación 5.1.

5.3. Resultados

A continuación se presentan los resultados obtenidos de la selección de variables.

5.3.1. Selección de variables

A pesar de tener un algoritmo que permite elegir las variables que más se adecuan al sistema original, éste no siempre elige las mismas. Los resultados arrojados

varían de acuerdo a los parámetros del SOM (dimensión de la malla de neuronas, radio y tasa de aprendizaje iniciales y número de épocas). A pesar de esto, existen similitudes en cada experimentación y cada combinación de parámetros, dando así una idea de qué variables son más importantes que otras.

La Tabla 5.2 y Figura 5.2 muestran lo anterior. Existen cinco variables ($d(MK)$, $d(YR)$, AC, CA y GT) que en ninguna experimentación se discriminaron, dos que su porcentaje de discriminación es bajo (AA y TC), dos que en toda experimentación fueron discriminadas (CG y GA), siete cuyo porcentaje de discriminación es superior al 80 % ($C+G$, AG, CC, GC, GG, TA y TG) y el resto de las variables (cuatro: $d(WS)$, AT, CT y TT) cuyo porcentaje de discriminación ronda entre el 43 % y 74 %.

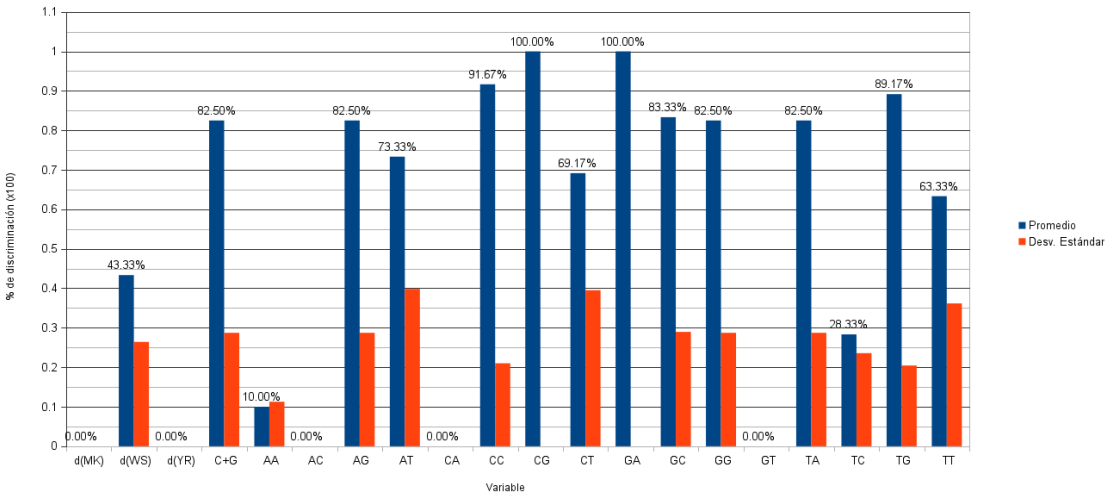


Figura 5.2: Porcentaje de discriminación de cada variable. Las barras azules muestran qué porcentaje de discriminación obtuvo cada variable. Las barras naranjas muestran la desviación estándar.

Las Figuras 5.3 y 5.4 muestran la distribución de los datos de la Tabla 5.2, esto es, visualizan la cantidad de veces que una variable fue discriminada de acuerdo a la combinación de parámetros utilizada en la experimentación. La primera muestra el conteo mientras que la segunda el porcentaje de ejecuciones totales, 120, dando como resultado el promedio descrito en la Taba 5.2 y Figura 5.2.

Con los datos anteriores se puede obtener un nuevo conjunto de datos, cada ejemplar con menor dimensión al original, basándose en un porcentaje máximo permitido. Para fines de comparación se eligen aquellos que cumplen con variables discriminadas menos de 85, 80, 75 y 0 %.

5.3.2. Clasificación

Los elementos de entrada describen propiedades de las cadenas de DNA que permiten cuantificar características termodinámicas y de estructura, las que varían

	d(MK)	d(WS)	d(YR)	C+G	AA	AC	AG	AT	CA	CC
C1	0 (00.00 %)	0 (00.00 %)	0 (00.00 %)	9 (60.00 %)	0 (00.00 %)	0 (00.00 %)	9 (60.00 %)	0 (00.00 %)	0 (00.00 %)	15 (100.00 %)
C2	0 (00.00 %)	2 (13.33 %)	0 (00.00 %)	3 (20.00 %)	0 (00.00 %)	0 (00.00 %)	3 (20.00 %)	3 (20.00 %)	0 (00.00 %)	6 (40.00 %)
C3	0 (00.00 %)	11 (73.33 %)	0 (00.00 %)	15 (100.00 %)	0 (00.00 %)	0 (00.00 %)	15 (100.00 %)	13 (86.67 %)	0 (00.00 %)	15 (100.00 %)
C4	0 (00.00 %)	4 (26.67 %)	0 (00.00 %)	15 (100.00 %)	4 (26.67 %)	0 (00.00 %)	15 (100.00 %)	15 (100.00 %)	0 (00.00 %)	15 (100.00 %)
C5	0 (00.00 %)	9 (60.00 %)	0 (00.00 %)	15 (100.00 %)	3 (20.00 %)	0 (00.00 %)	15 (100.00 %)	15 (100.00 %)	0 (00.00 %)	15 (100.00 %)
C6	0 (00.00 %)	9 (60.00 %)	0 (00.00 %)	13 (86.67 %)	0 (00.00 %)	0 (00.00 %)	13 (86.67 %)	13 (86.67 %)	0 (00.00 %)	14 (93.33 %)
C7	0 (00.00 %)	8 (53.33 %)	0 (00.00 %)	15 (100.00 %)	2 (13.33 %)	0 (00.00 %)	15 (100.00 %)	15 (100.00 %)	0 (00.00 %)	15 (100.00 %)
C8	0 (00.00 %)	9 (60.00 %)	0 (00.00 %)	14 (93.33 %)	3 (20.00 %)	0 (00.00 %)	14 (93.33 %)	14 (93.33 %)	0 (00.00 %)	15 (100.00 %)
Promedio	**0 (00.00 %)**	**6.5 (43.33 %)**	**0 (00.00 %)**	**12.375 (82.50 %)**	**1.5 (10.00 %)**	**0 (00.00 %)**	**12.375 (82.50 %)**	**11 (73.33 %)**	**0 (00.00 %)**	**13.75 (91.67 %)**
Desv. Est.	**00.00 (00.00)**	**03.96 (26.43)**	**00.00 (00.00)**	**04.31 (28.72)**	**01.69 (11.27)**	**00.00 (00.00)**	**04.31 (28.72)**	**05.98 (39.84)**	**00.00 (00.00)**	**03.15 (21.01)**

	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT
C1	15 (100.00 %)	0 (00.00 %)	15 (100.00 %)	9 (60.00 %)	9 (60.00 %)	0 (00.00 %)	9 (60.00 %)	0 (00.00 %)	13 (86.67 %)	0 (00.00 %)
C2	15 (100.00 %)	2 (13.33 %)	15 (100.00 %)	3 (20.00 %)	3 (20.00 %)	0 (00.00 %)	3 (20.00 %)	2 (13.33 %)	6 (40.00 %)	2 (13.33 %)
C3	15 (100.00 %)	13 (86.67 %)	15 (100.00 %)	15 (100.00 %)	15 (100.00 %)	0 (00.00 %)	15 (100.00 %)	11 (73.33 %)	15 (100.00 %)	11 (73.33 %)
C4	15 (100.00 %)	15 (100.00 %)	15 (100.00 %)	15 (100.00 %)	15 (100.00 %)	0 (00.00 %)	15 (100.00 %)	4 (26.67 %)	15 (100.00 %)	15 (100.00 %)
C5	15 (100.00 %)	15 (100.00 %)	15 (100.00 %)	15 (100.00 %)	15 (100.00 %)	0 (00.00 %)	15 (100.00 %)	7 (46.67 %)	15 (100.00 %)	13 (86.67 %)
C6	15 (100.00 %)	12 (80.00 %)	15 (100.00 %)	14 (93.33 %)	13 (86.67 %)	0 (00.00 %)	13 (86.67 %)	1 (06.67 %)	14 (93.33 %)	12 (80.00 %)
C7	15 (100.00 %)	12 (80.00 %)	15 (100.00 %)	15 (100.00 %)	15 (100.00 %)	0 (00.00 %)	15 (100.00 %)	4 (26.67 %)	15 (100.00 %)	12 (80.00 %)
C8	15 (100.00 %)	14 (93.33 %)	15 (100.00 %)	14 (93.33 %)	14 (93.33 %)	0 (00.00 %)	14 (93.33 %)	5 (33.33 %)	14 (93.33 %)	11 (73.33 %)
Promedio	**15 (100.00 %)**	**10.375 (69.17 %)**	**15 (100.00 %)**	**12.5 (83.33 %)**	**12.375 (82.50 %)**	**0 (00.00 %)**	**12.375 (82.50 %)**	**4.25 (28.33 %)**	**13.375 (89.17 %)**	**9.5 (63.33 %)**
Desv. Est.	**00.00 (00.00)**	**05.93 (39.51)**	**00.00 (00.00)**	**04.34 (28.95)**	**04.31 (28.72)**	**00.00 (00.00)**	**04.31 (28.72)**	**03.54 (23.57)**	**03.07 (20.45)**	**05.42 (36.17)**

Tabla 5.2: Resultados de los experimentos. Los datos presentados son el número de veces (porcentaje) que una variable es discriminada.

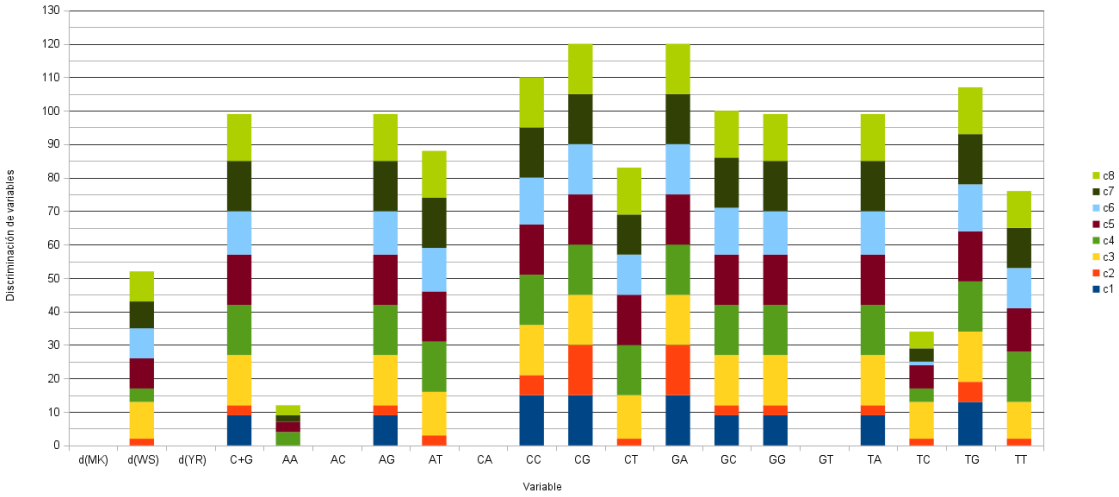


Figura 5.3: Discriminación de variables por combinación de parámetros. Cada variable muestra el número de veces que fue discriminada de acuerdo a la combinación de parámetros. 120 es el número de total de experimentos realizados (por las 8 combinaciones), 15 por cada combinación.

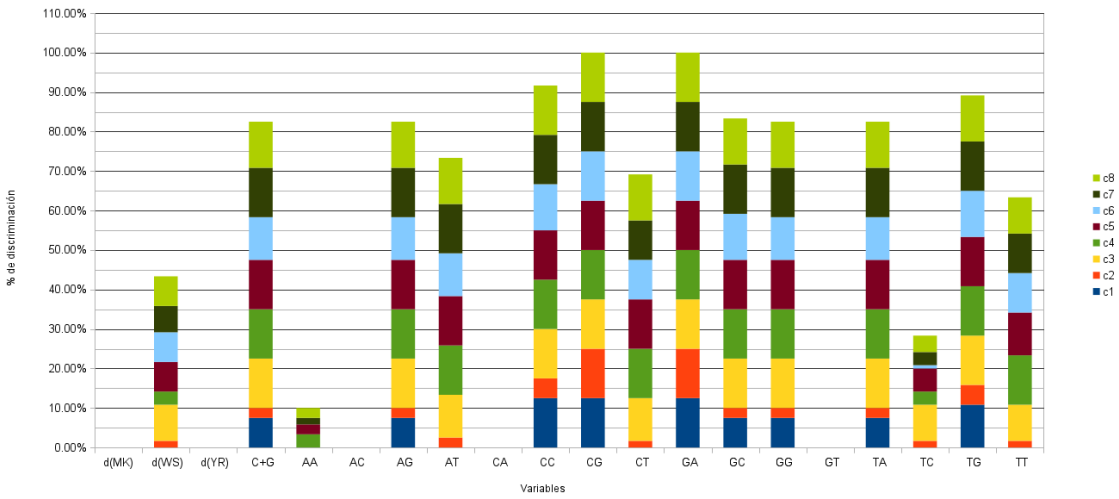


Figura 5.4: Porcentaje de discriminación de variables por combinación de parámetros. Cada variable muestra en la gráfica el porcentaje de discriminación de acuerdo a los parámetros utilizados.

de especie en especie. Al aplicar el algoritmo de Kohonen a estos datos, no se emplea el uso de algún maestro que oriente a las neuronas a responder a cada elemento, sino que por sí sólo puede encontrar las similitudes.

Conforme se realiza el entrenamiento, las neuronas aprenden a responder a ciertas entradas, por lo que el mapa neuronal se va ordenando de acuerdo a ellas y así encuentra un orden. Neuronas cercanas entre sí indican respuestas a entradas similares por lo que el orden topológico se establece de esta manera.

Para poder visualizar los resultados se muestran las gráficas con etiquetado de neuronas y la matriz unificada de distancias en diferentes épocas de entrena-

miento.

En la primer época de entrenamiento, el mapa de neuronas carece de un orden específico. La Figura 5.5b es un histograma que cuenta el número de vectores de entrada que estimulan a la neurona; entre más oscuro es el color, mayor es el número de muestras que la estimulan. Sólo unas pocas neuronas responden ante las entradas por lo que la Figura 5.5a muestra sólo algunas neuronas con etiquetado.

Para la época 450 (Figura 5.6) la cantidad de neuronas que responden a los vectores de entrada es mayor comparando con la primera iteración. Además las neuronas etiquetadas con la misma clase aparecen en general dentro de la misma vecindad topológica (Figura 5.6a) y la matriz de distancias marca separación entre neuronas de diferentes clases e incluso de la misma.

El resultado final del entrenamiento se muestra en la Figura 5.7. Las neuronas muestran similitud en cuanto al número de entradas al que responden. El etiquetado de neuronas mediante el método de Kuri [37] indica a qué clase pertenece cada neurona. Neuronas vecinas topológicamente obtienen la misma clase. Esto es porque reciben estímulos de patrones de entrada similares. La matiz de distancias unificadas muestra las distancias entre neuronas vecinas dentro de la red de neuronas permitiendo observar qué tanto parecido existe entre ellas. Los colores claros indican grupos de neuronas similares mientras que los colores oscuros indican los límites de esos grupos. Esta matriz permite identificar *clusters* de los elementos de entrada a través de las neuronas mediante sus distancias. Elementos del mismo grupo tienen vectores de referencia cercanos, mientras que elementos de diferentes grupos, las distancias de los vectores de pesos son mayores.

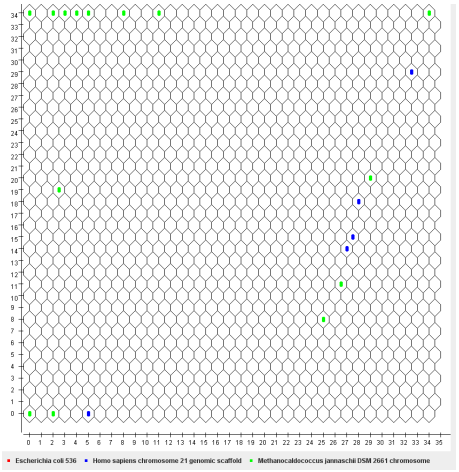
La clasificación al conjunto de datos sin variables dispensables tiene un proceso similar, el cual no se muestra. Únicamente se muestran los resultados obtenidos al término del entrenamiento en las Figuras 5.8, 5.9, 5.10 y 5.11.

5.4. Resumen

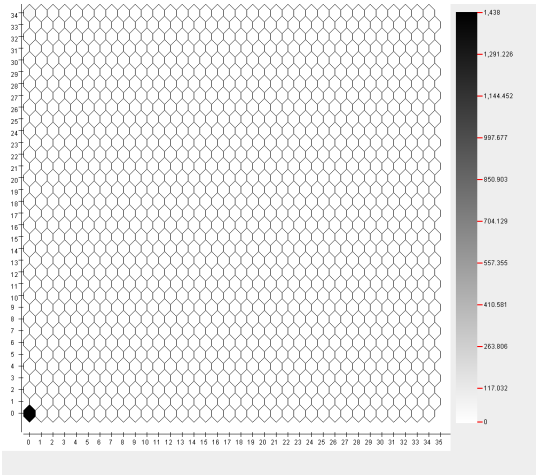
Las secuencias de DNA son codificadas de forma que puedan ser tratadas para su clasificación. Esta codificación nos da un total de 1500 vectores de dimensión 20, donde cada elemento del vector representa la medición de una codificación de un segmento de DNA. La información contenida en estos vectores no necesariamente es útil; hay variables que pueden agregar ruido, por lo que es necesario eliminarlas con el fin de tener resultados con la menor alteración posible. Para esto se hizo uso de reducción de variables.

La reducción de variables eliminó nueve variables con lo que la clasificación de datos se realiza con las once restantes. No obstante, se hace la clasificación con el conjunto de datos completo con el fin de comparar los resultados arrojados.

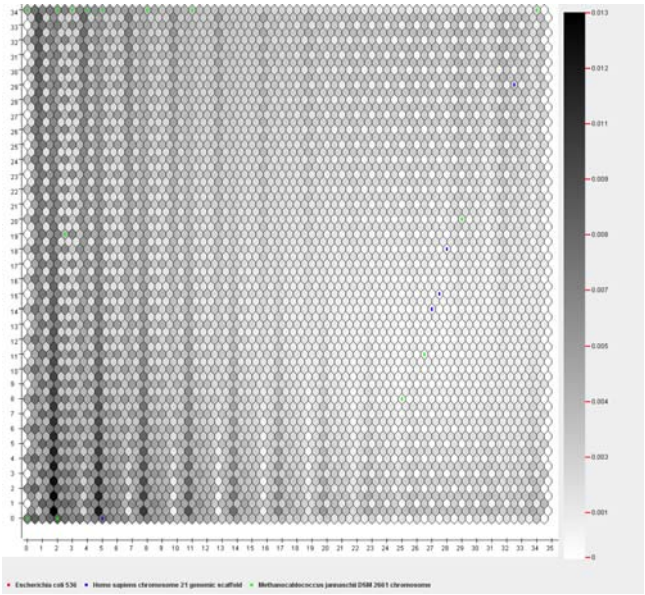
En cuanto a la clasificación, la red neuronal puede distinguir y agrupar los



(a) Etiquetas de las neuronas. Muestra a qué clase responde cada neurona.

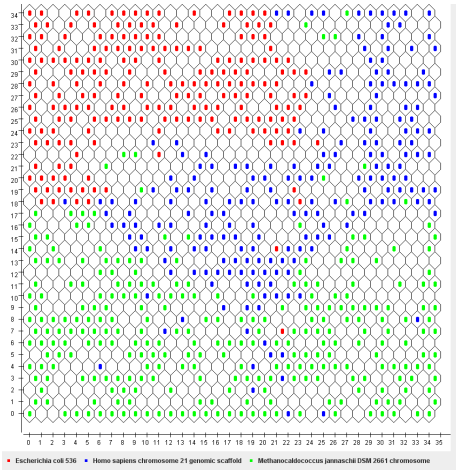


(b) Gráfica de frecuencias. Cada neurona muestra a cuántos vectores de entrada responde.

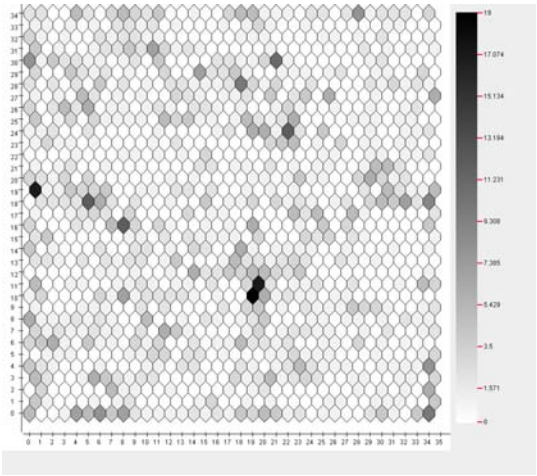


(c) *U-Matrix*. Muestra las distancias entre neuronas

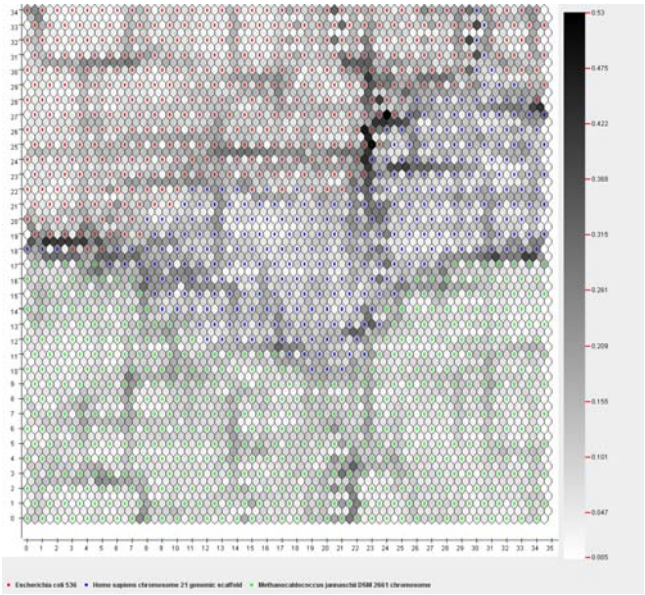
Figura 5.5: Primera época de entrenamiento



(a) Etiquetas de las neuronas. Muestra a qué clase responde cada neurona.

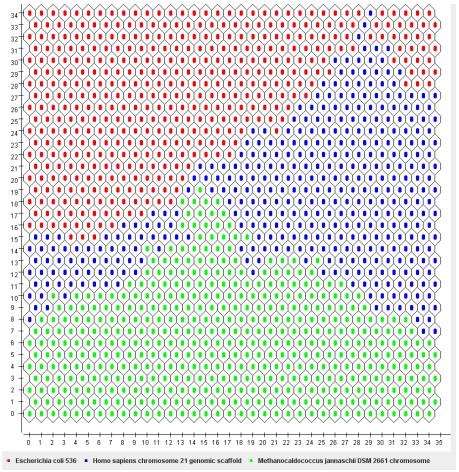


(b) Gráfica de frecuencias. Cada neurona muestra a cuántos vectores de entrada responde.

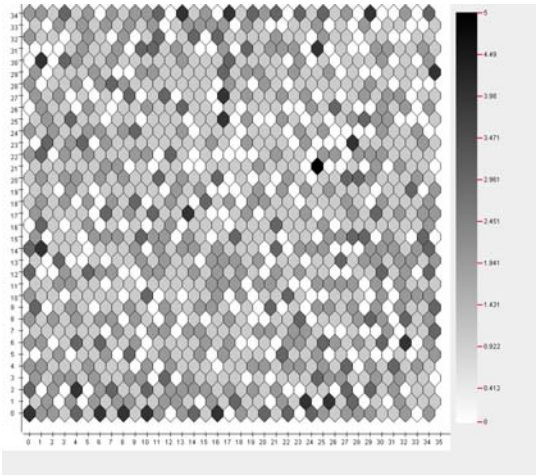


(c) *U-Matrix*. Muestra las distancias entre neuronas

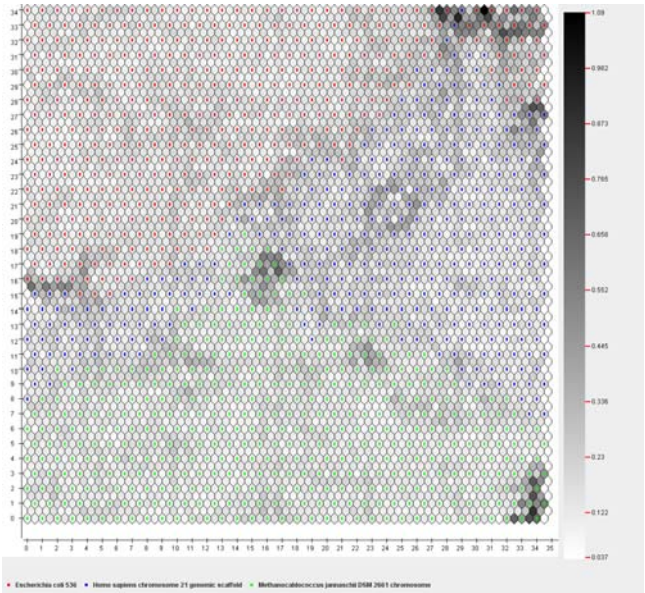
Figura 5.6: Época 450 de entrenamiento.



(a) Etiquetas de las neuronas. Muestra a qué clase responde cada neurona.

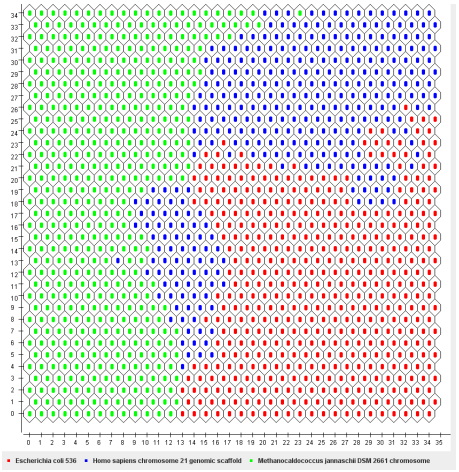


(b) Gráfica de frecuencias. Cada neurona muestra a cuántos vectores de entrada responde.

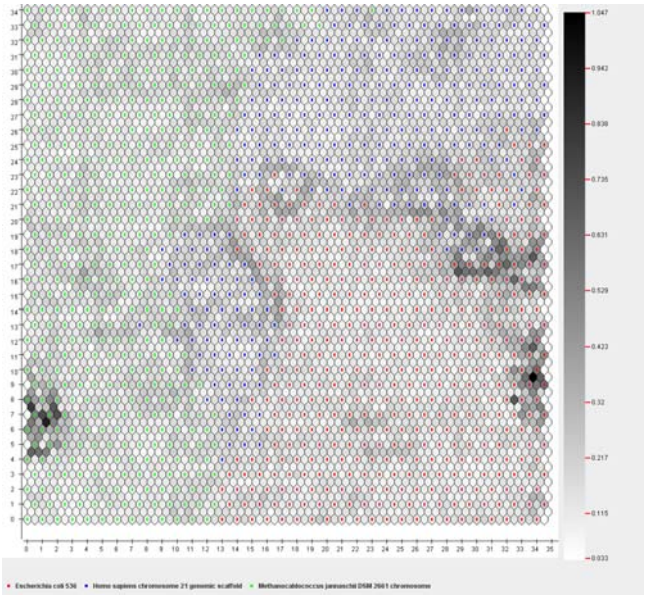


(c) *U-Matrix*. Muestra las distancias entre neuronas

Figura 5.7: Resultados del entrenamiento.



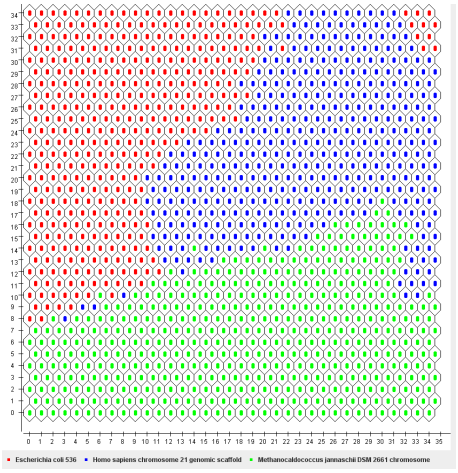
(a) Etiquetas de las neuronas. Muestra a qué clase responde cada neurona.



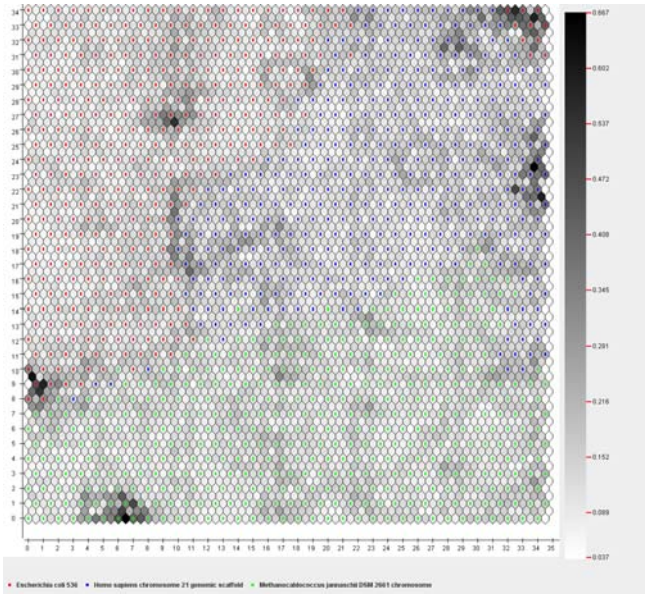
(b) *U-Matrix*. Muestra las distancias entre neuronas

Figura 5.8: Resultado del entrenamiento del conjunto de datos sin las variables que fueron discriminadas más del 85 % de veces.

elementos de acuerdo a sus características. El etiquetado del mapa de neuronas muestra lo anterior manteniendo neuronas vecinas topológicamente con la misma clase.

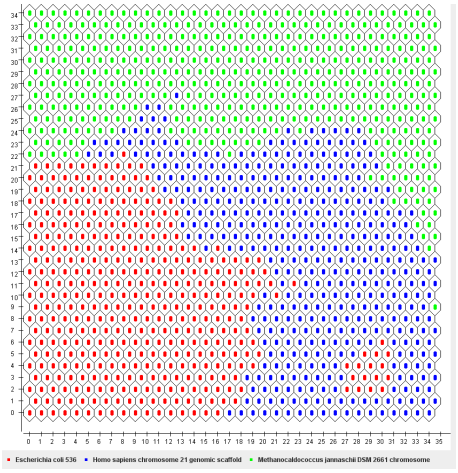


(a) Etiquetas de las neuronas. Muestra a qué clase responde cada neurona.

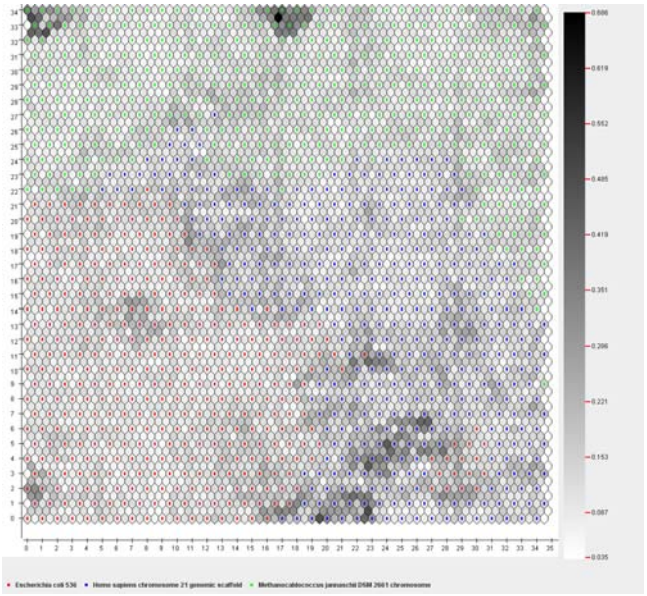


(b) *U-Matrix*. Muestra las distancias entre neuronas

Figura 5.9: Resultado del entrenamiento del conjunto de datos sin las variables que fueron discriminadas más del 80 % de veces.

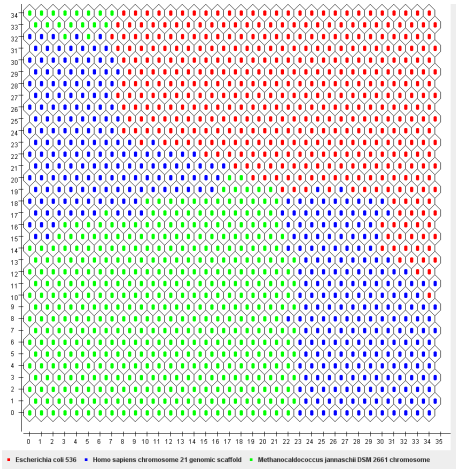


(a) Etiquetas de las neuronas. Muestra a qué clase responde cada neurona.

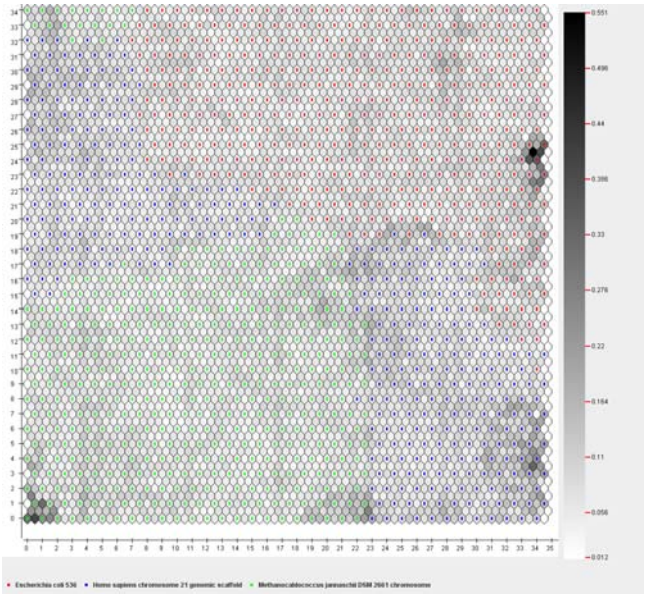


(b) *U-Matrix*. Muestra las distancias entre neuronas

Figura 5.10: Resultado del entrenamiento del conjunto de datos sin las variables que fueron discriminadas más del 70 % de veces.



(a) Etiquetas de las neuronas. Muestra a qué clase responde cada neurona.



(b) *U-Matrix*. Muestra las distancias entre neuronas

Figura 5.11: Resultado del entrenamiento del conjunto de datos manteniendo únicamente aquellas que nunca fueron discriminadas.

Capítulo 6

Conclusiones

La Figura 5.2 muestra el porcentaje de discriminación que obtiene cada variable. Entre más alta la columna, menos peso informativo tiene la misma. Destaca el hecho de que dos de las tres variables ($d(YR), d(MK)$) sean indispensables y la otra ($d(WS)$) no tanto. Esto no es de extrañar dado que una vez que una secuencia de DNA se traduce a dos secuencias binarias, la tercera queda determinada [44]. Sin embargo, la relación funcional entre dichas variables no ha sido aún determinada.

La estructura de dímeros del DNA determina, entre otras cosas, la energía de enlace del dúplex [9] que mantiene la estabilidad de la cadena de DNA. Dado que existen solamente diez interacciones independientes entre dímeros consecutivos y sus contrapartes bajo la interacción Watson-Crick, los resultados de la Figura 5.2 son congruentes con este hecho. Diez dímeros deben bastar para discriminar adecuadamente entre diferentes secuencias de DNA y en la misma figura se aprecia que bajo ciertos porcentajes de discriminación, los dímeros mostrados en la Tabla 6.1 tienen las proporciones más altas de discriminación. El por qué precisamente éstos y no otros queda fuera del alcance de este trabajo.

Con base en los datos obtenidos por el preprocesamiento de los datos detallado en el Capítulo 5 se procede a aplicar mapas autoorganizados en el conjunto de datos con todas las variables y en los conjuntos de datos reducidos

.

También se aplicó, en ánimo exploratorio, el método de componentes principales, utilizado para el análisis y reducción de variables del conjunto de datos. No obstante, sus resultados no son presentados pues no arrojan información relevante

Porcentaje	**Dímeros discriminados**
85	CC, CG, GA, TG
80	$AG, CC, CG, GA, GC, GG, TA, TG$
70	$AG, AT, CC, CG, GA, GC, GG, TA, TG$
0	$AA, AG, AT, CC, CG, CT, GA, GC, GG, TA, TC, TG, TT$

Tabla 6.1: Dímeros discriminados de acuerdo al porcentaje de discriminación.

ni interpretable para el tema de estudio tratado.

Como se mostró en las Figuras 5.7a, 5.8a, 5.9a, 5.10a y 5.11a, éstas muestran la clasificación de las neuronas de acuerdo a las clases de los vectores de entrada. Neuronas vecinas responden a estímulos semejantes por lo que muestran propiedades termodínamicas y estructurales semejantes. El etiquetado de neuronas refuerza este hecho. La matriz unificada de distancias *U-Matrix* mostradas en las Figuras 5.7c, 5.8b, 5.9b, 5.10b y 5.11b muestran las distancias entre neuronas vecinas. Mediante este método podemos analizar cómo agrupar la red a los elementos de entrada mediante las neuronas de la red y las distancias entre ellas. Mayores distancias de pesos entre neuronas (colores oscuros) indican límites entre grupos, mientras que menores distancias (colores claros) indican los grupos. En cada figura se muestran en diferentes partes colores muy oscuros indicando distancias grandes entre neuronas, pertenecientes a diferentes especies. Esto es razonable debido a la diferencia entre las propiedades de las subcadenas de DNA codificadas a las que responden dichas neuronas, pues deben compartir similitudes estructuras y termodinámicas entre ellas. No obstante también existen distancias cercanas (o colores claros) entre neuronas etiquetadas con diferentes especies o distancias grandes (colores oscuros) entre neuronas etiquetadas con la misma especie lo que indica existencia de similitudes o disimilitudes entre diferentes o las mismas especies. Éstas han quedado fuera del alcance del trabajo para un posterior estudio que además permita tener subclasificaciones.

La utilización de clases en el etiquetado de neuronas ayuda a discernir entre los diferentes *clusters* definidos por la red neuronal, por lo que una métrica que unifique *U-Matrix* y el etiquetado valdría la pena estudiar.

La clasificación realizada está limitada a agrupar las especies analizadas a la categoría taxonómica más alta, llamada dominio, dentro de la clasificación biológica. Los resultados muestran una separación entre los dos dominios biológicos: el *eukaryota*, al que pertenece el *homo sapiens* y el *prokaryota* al que pertenecen las dos bacterias. En ambos tipos de conjuntos de datos, el que contiene todas las variables y los que no contiene las variables discriminadas, la separación de clases es notoria, pese a que existen puntos "invasores" de clases (especies) no pertenecientes a otra. Pero para tener el conocimiento de qué clasificación es mejor, se recurre a las métricas de errores topológicos[32] y cuantitativos[3] mostrados en las Figuras 6.1 y 6.2 respectivamente.

En ambas figuras, la línea roja muestra los errores del conjunto de datos con todas las variables a lo largo del entrenamiento mientras que las líneas verde, azul, magenta y azul claro las del conjunto de datos sin variables discriminadas (aquellas por debajo o igual al 85, 80, 70 y 0 % respectivamente). La línea amarilla representa un conjunto de datos con únicamente dos variables (elegidas al azar): $C + G$ y TT. Se puede apreciar que a medida que se tienen menor cantidad de variables, los errores cuantitativos disminuyen (Figura 6.2), sin embargo, no sucede lo mismo con los errores topológicos (Figura 6.1) donde el error topológico mostrado por los datos con las variables $C + G$ y TT es mayor al resto

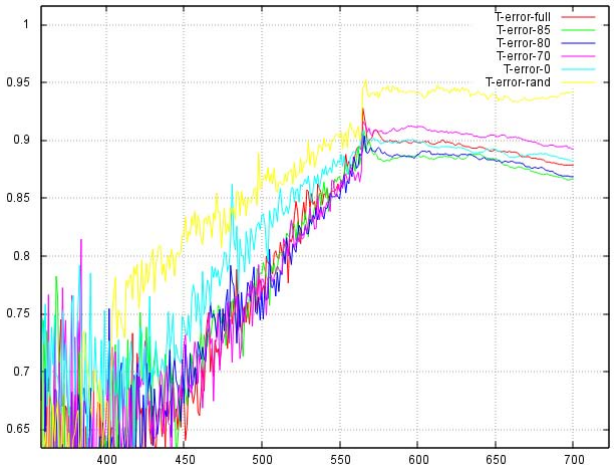


Figura 6.1: Error topológico de los dos mapas autoorganizados. El eje de las X representa las iteraciones del entrenamiento mientras que el eje de las Y representa el error topológico . La línea roja da a conocer el comportamiento del error utilizando todas las variables, mientras que el resto muestran lo mismo pero con diferentes subconjuntos de variables.

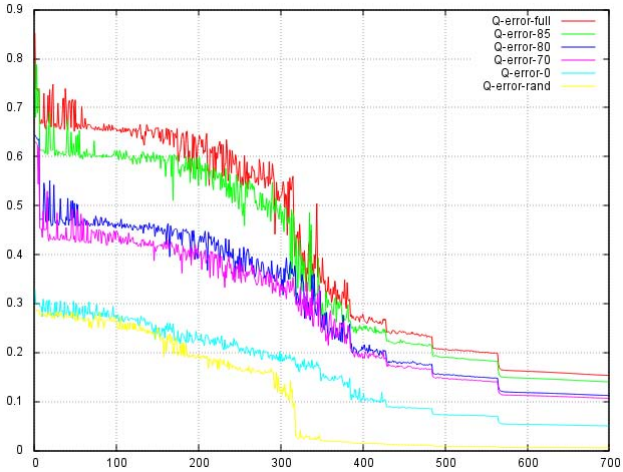


Figura 6.2: Error cuantitativo de los dos mapas autoorganizados. El eje de las X representa las iteraciones del entrenamiento mientras que el eje de las Y representa el error cuantitativo. La línea roja da a conocer el comportamiento del error utilizando todas las variables, mientras que el resto muestran lo mismo pero con diferentes subconjuntos de variables.

mostrando un orden topológico menor al resto. La red neuronal entrenada con un máximo porcentaje de discriminación del 85 y 80 % muestran el mejor ordenamiento topológico del mapa. Tomando en cuenta ambas gráficas, podemos concluir que el conjunto de datos con mejor calidad es el que tiene un máximo de discriminación del 80 %, siendo las variables que mejor describen a los datos $d(MK), d(WS), d(YR), AG, CC, CG, GA, GC, GG, TA, TG$. No obstante, el estudio de otras métricas de comparación entre mapas neuronales que permitan obtener una mejor elección entre ellos, se deja como trabajo futuro.

Bibliografía

[1] Alahakoon, D., Halgamuge, S., and Srinivasan, B. Dynamic self-organizing maps with controlled growth for knowledge discovery. *Neural Networks, IEEE Transactions on 11*, 3 (May 2000), 601–614.

[2] Anderson, T. W. *An Introduction to Multivariate Analysis*, 3ra ed. John Wiley & Sons, 2003.

[3] Arsuaga-Uriarte, E., and Díaz-Martín, F. Topology preservation in som. In *International Journal of Applied Mathematics and Computer Sciences* (2005), W.A.S.E.T., pp. 19–22.

[4] Baofeng, G., and Nixo, M. Gait feature subset selection by mutual information. *IEEE Transactions on Systems, Man and Cybernetics, Part A 39* (2008), 36–46.

[5] Battiti, R. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks 5*, 4 (1994), 537–550.

[6] Beale, R., and Jackson, T. *Neural Computing: An Introduction*. IOP Publishing, 1990.

[7] Benabdeslem, K., and Lebbah, M. Feature selection for self-organizing map. In *Information Technology Interfaces, 2007. ITI 2007. 29th International Conference on* (June 2007), pp. 45–50.

[8] Blackmore, J., and Miikkulainen, R. Incremental grid growing: encoding high-dimensional structure into a two-dimensional feature map. In *Neural Networks, 1993., IEEE International Conference on* (1993), pp. 450–455 vol.1.

[9] Breslauer, K. J., Frank, R., Blocker, H., and Marky, L. A. Predicting DNA Duplex Stability from the Base Sequence. *Proceedings of The National Academy of Sciences 83* (1986), 3746–3750.

[10] Calderón, C. Redes neuronales para la detección de genes foráneos, 2004.

[11] Cang, S., and Yu, H. Mutual information based input feature selection for classification problems. *Decision Support Systems*, 54 (2012), 691–698.

[12] Chow, T. W., and Huang, D. Data reduction for pattern recognition and data analysis. In *Computational Intelligence: A Compendium Studies in Computational Intelligence*, vol. 115. 2008, pp. 81–109.

[13] Cover, T., and Thomas, J. *Elements of Information Theory*, 2da ed. John Wiley and Sons, INC., 2006.

[14] Dy, J. Unsupervised feature selection. In *Computational Methods for Feature Selection*, H. Lui and H. Motoda, Eds. CRC Press, 2007.

[15] Estévez, P., Tesmer, M., and Pérez, C. Normalized mutual information feature selection. *IEEE Transactions on Neural Networks 2*, 20 (2009), 189–201.

[16] Freeman, J. A., and Skapura, D. M. *Neural Networks. Algorithms, Applications and Programming Techniques.* Addison-Wesley, 1991.

[17] Fritzke, B. Growing cell structures - a self-organizing network for unsupervised and supervised learning. *Neural Networks 7* (1993), 1441–1460.

[18] Fritzke, B. A growing neural gas network learns topologies. In *Advances in Neural Information Processing Systems 7* (1995), MIT Press, pp. 625–632.

[19] Fritzke, B. Growing self-organizing networks - why? In *In ESANN'96: European Symposium on Artificial Neural Networks* (1996), Publishers, pp. 61–72.

[20] Fritzke, B. Growing self-organizing networks—history, status quo, and perspectives. In *Kohonen Maps*, E. Oja and S. Kaski, Eds. Elsevier Science B.V., Amsterdam, 1999, pp. 131 – 144.

[21] Gatignon, H. *Statistical Analysis of Management Data*, 3da ed. Springer, 2014.

[22] Gersho, A., and Gray, R. M. *Vector Quantization and Signal Compression.* Kluwer Academic Publishers, Norwell, MA, USA, 1991.

[23] Guyon, I., and Elisseeff, A. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research 3* (2003), 1289–1305.

[24] Guyon, I., and Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res. 3* (Mar. 2003), 1157–1182.

[25] Hand, D. J., Smyth, P., and Mannila, H. *Principles of Data Mining.* MIT Press, Cambridge, MA, USA, 2001.

[26] Haykin, S. *Neural Networks. A comprehensive Foundation.* Prentice Hall International, 1999.

[27] Hertz, J., Krogh, A., and Palmer, R. *Introduction to the Theory of Neural Computation.* Addison-Wesly, 1991.

[28] Honkela, T., Kaski, S., Lagus, K., and Kohonen, T. Websom - self-organizing maps of document collections. In *Neurocomputing* (1997), pp. 101–117.

[29] Horton, H., and Moran, L. e. a. *Principles of Biochemistry*, 4ta ed. Pearson Prentice-Hall, 2006.

[30] Jain, A. K., Murty, M. N., and Flynn, P. J. Data clustering: A review. *ACM Comput. Surv. 31*, 3 (1999), 264–323.

[31] Jiawei, H., and Kamber, M. *Data Mining: Concepts and Techniques*, 2da ed. Morgan Kauffmann, 2006.

[32] Kiviluoto, K. Topology preservation in self-organizing maps. In *Neural Networks, 1996., IEEE International Conference on* (Jun 1996), vol. 1, pp. 294–299 vol.1.

[33] Kohonen, T. The ñeural'phonetic typewriter. *Computer 21*, 3 (March 1988), 11–22.

[34] Kohonen, T. The self organizig map. *Proceedings of the IEEE 78*, 9 (1990), 1464–1480.

[35] Kohonen, T. *Self-Organizig Maps*, 3ra ed. Spring-Verlang, 2000.

[36] Kohonen, T., Kaski, S., Lagus, K., Salojarvi, J., Paatero, V., and Saarela, A. Self organization of a massive document collection. *IEEE Transactions on Neural Networks 11*, 3 (2000), 574–585.

[37] Kuri-Morales, A. Automatic clustering with self-organizing maps and genetic algorithms. `https://www.researchgate.net/publication/255599449_Automatic_Clustering_with_Self-Organizing_Maps_and_Genetic_Algorithms_II_an_Improved_Approach`.

[38] Kwak, N., and Choi, C. Input feature selection by mutual information based on parzen window. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 12 (2002), 1667–1671.

[39] Kwak, N., and Choi, C. Input feature selection for classification problems. *IEEE Transactions on Neural Networks*, 13 (2002), 143–159.

[40] Ladha, L., and Deepa, T. Feature selection methods and algorithms. *International Journal on Computer Science and Engineering 3*, 5 (2011), 1787–1797.

[41] Marill, T., and Green, D. M. On the effectiveness of receptors in recognition systems. *Information Theory, IEEE Transactions on 9*, 1 (1963), 11–17.

[42] Martinetz, T. Competitive hebbian learning rule forms perfectly topology preserving maps. In *ICANN '93*, S. Gielen and B. Kappen, Eds. Springer London, 1993, pp. 427–434.

[43] Martinetz, T. M., and Schulten, K. J. A "neural gas" network learns topologies. In *Proceedings of the International Conference on Artificial Neural Networks 1991* (Espoo, Finland) (1991), T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas, Eds., Amsterdam; New York: North-Holland, pp. 397–402.

[44] Miramontes, P., Medrano, L., Cerpa, C., Cedergren, R., Ferbeyre, G., and Cocho, G. Structural and thermodynamic properties of dna uncover different evolutionary histories. *Journal of Molecular Evolution 40*, 6 (1995), 698–704.

[45] Miramontes Vidal, P. *Un Esquema de Autómata Celular como Modelo Matemático de la Evolución de los Ácidos Nucleicos*. PhD thesis, Universidad Nacional Autónoma de México, 1992.

[46] Neme, A., and Miramontes, P. Self-organizing map formation with a selectively refractory neighborhood. *Neural Processing Letters 39*, 1 (2014), 1–24.

[47] Nova, D., and Estévez, P. A review of learning vector quantization classifiers. *Neural Computing and Applications 25*, 3-4 (2014), 511–524.

[48] Ontrup, J., and Ritter, H. Hyperbolic self-organizing maps for semantic navigation, 2001.

[49] Peng, H., Long, F., and Ding, C. Feature selection based on mutual information: criteria of max-dependency, max-relevance and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence 8*, 27 (2004), 1226–1238.

[50] Pudil, P., Novovičová, J., and Kittler, J. Floating search methods infeature selection. *Pattern Recognition Letters 15*, 11 (1994), 1119–1125.

[51] Rauber, A., Merkl, D., and Dittenbach, M. The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data. *Neural Networks, IEEE Transactions on 13*, 6 (Nov 2002), 1331–1341.

[52] Raymer, M., Punch, W., Goodman, E., Kuhn, L., and Jain, A. Dimensionality reduction using genetic algorithms. *Evolutionary Computation, IEEE Transactions on 4*, 2 (Jul 2000), 164–171.

[53] Ritter, H. Self-organizing maps on non-euclidean spaces. In *Kohonen Maps* (1999), Elsevier, pp. 97–108.

[54] Rojas, R. *Neural Networks. A Systematic Introduction*. Springer, 1996.

[55] Ross, S. *A First Course of Probability*, 8va ed. Prentice Hall, 2010.

[56] Rumelhart, D., and Zipser, D. Feature discovery by competitive learning. *Cognitive Science 9* (1985), 75 – 112.

[57] Shlens, J. *A Tutorial on Principal Component Analysis*, 2005.

[58] Stearns, S. D. On selecting features for pattern classifiers. In *Proceedings of the 3rd International Conference on Pattern Recognition (ICPR 1976)* (1976), pp. 71–75.

[59] Stracuzzi, D. Randomized feature selection. In *Computational Methods for Feature Selection*, Lui and Motoda, Eds. CRC Press, 2007.

[60] u. Bauer, H., and Villmann, T. Growing a hypercubical output space in a self-organizing feature map. *IEEE Transactions on Neural Networks 8* (1995), 218–226.

[61] Ultsch, A., and Siemon, H. P. Kohonen's selforganizing feature maps for exploratory data analysis. In *In Proceeding of International Neural Network Conference (INNC'90), Kluwer Academic Press* (Dordrecht, Netherlands, 1990), pp. 305–308.

[62] Van Hulle, M. Self-organizing maps. In *Handbook of Natural Computing*, G. Rozenberg, T. Bäck, and J. Kok, Eds. Springer Berlin Heidelberg, 2012, pp. 585–622.

[63] Vergara, J., and Estévez, P. A review of feature selection methods based on mutual information. *Neural Computing and Applications 24*, 1 (2014), 175–186.

[64] Vinh, L., Lee, S., Park, Y.-T., and d'Auriol, B. A novel feature selection method based on normalized mutual information. *Applied Intelligence 37*, 1 (2012), 100–120.

[65] Voet, D., and Voet, J. *Biochemistry*, 4ta ed. John Wiley and Sons, INC., 2011.

[66] Wang, J., Wu, L., Kong, J., Li, Y., and Zhang, B. Maximum weight and minimum redundancy: A novel framework for feature subset selection. *Pattern Recognition 46*, 6 (2013), 1616 – 1627.

[67] Watson, J. e. a. *Molecular Biology of the Gene*, 7ma ed. Pearson, 2014.

[68] Whitney, A. A direct method of nonparametric measurement selection. *Computers, IEEE Transactions on C-20*, 9 (1971), 1100–1103.

[69] Xu, R., and Wunsh, D. Survey on clustering algorithms. *IEEE Transactions on Neural Networks 16*, 3 (2005), 645–678.

[70] Yang, H., and Moody, J. Feature selection based on joint mutual information. *Advances in intelligent data analysis, proceedings of international ICSC symposium* (1999), 22–25.

[71] Yin, H. The self-orginizing maps: Background, theories, extensions and applications. In *Studies in Computational Intelligence*. Springer-Verlang, 2008, pp. 715–762.

[72] Zheng, H., and Wu, H. Gene-centric association analysis for the correlation between the guanine-cytosine content levels and temperature range conditions of prokaryotic species. *BMC Bioinformatics 11*, Suppl 11 (2010).

[73] Zheng, Y., and Kwoh, C. K. A feature subset selection method based on high-dimensional mutual information. *Entropy 13*, 4 (2011), 860–901.