Identificación de Patrones en Antibiogramas Mediante Técnicas de Aprendizaje Automático

Lic. Carlos A. González Rubio Gavarain
Instituto Tecnológico de La Paz
La Paz B.C.S., México C.P. 23080
Email: gonzalezMSC1705@itlp.edu.mx

Dr. Marco Antonio Castro Liera
Instituto Tecnológico de La Paz
La Paz B.C.S., México C.P. 23080
Email: mcastro@itlp.edu.mx

***Resumen*—Se detalla la creación de una solución informática para facilitar la descripción y visualización de los patrones subyacentes en los datos mixtos multidimensionales de los antibiogramas.**

El enfoque de desarrollo utilizado es conocido como "descubrimiento de conocimiento en bases de datos" cuyo objetivo es identificar patrones válidos y novedosos a partir de datos existentes, mismos que son suministrados en forma de ejemplos entrenantes al algoritmo de mapa auto-organizado generalizado, utilizado durante la etapa de minado de datos para obtener un modelo que mapea datos de alta dimensión en una malla bidimensional que preserva las propiedades topológicas de los datos originales.

***Palabras clave*—Antibiogramas, Farmacorresistencia, Aprendizaje no supervisado, Minería de datos, Mapas Auto-Organizados, Distancia Jerárquica.**

I. Introducción

El antibiograma es una prueba microbiológica que se realiza para determinar la susceptibilidad de una bacteria a un grupo de antibióticos, misma que es realizada en las unidades hospitalarias para brindar opciones terapéuticas en el tratamiento de los pacientes.

Los antibiogramas se realizan en el momento del ingreso de un paciente al nosocomio para verificar que no tenga o esté incubando algún microorganismo, a los cinco días después de su ingreso o si presenta picos febriles.

Las IAAS (Infección Asociada a la Atención de la Salud) es un evento adverso durante la prestación de atención sanitaria y son de especial preocupación debido al uso inherente de antibióticos que genera presión selectiva creando microorganismos difíciles de tratar por el fenómeno conocido como farmacorresistencia.

La farmacorresistencia en unidades hospitalarias es un tema que ya ha sido investigado adoptando enfoques estadísticos, sin embargo, la falta de consenso y estructura hacen que se trate de un tópico poco formalizado.

La epidemiología es el estudio de la distribución y los determinantes de los estados o acontecimientos relacionados con la salud en poblaciones específicas y la aplicación de este estudio al control de los problemas sanitarios [1].

El presente trabajo describe la problemática y las soluciones llevadas a cabo durante el desarrollo de un programa informático que permita conocer el número, distribución y similitud de los microorganismos aislados en las unidades hospitalarias de acuerdo a los campos contenidos en los antibiogramas.

II. El proceso de descubrimiento de conocimiento en bases de datos

Teniendo como objetivo la generación de un modelo a partir de las características extraídas de los antibiogramas existentes se utilizó un proceso de desarrollo diseñado para transformar datos en conocimiento conocido como KDD [2] (por las siglas en inglés de *Knowledge Discovery in Databases*) donde se encapsulan las responsabilidades en un conjunto de pasos que generan una salida a partir de los datos procesados de la entrada del paso anterior como se muestra en la figura 1.

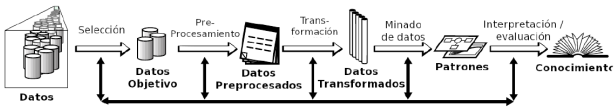

Figura 1: Pasos del proceso de descubrimiento de conocimiento en bases de datos

II-A. Creación de la base de datos

Con la finalidad de poder utilizar los datos de los antibiogramas se creó una base de datos relancional *SQL normalizada* (proceso para evitar la redundancia en los datos) utilizando el sistema de gestión de bases datos *MariaDB*.

II-B. Extracción, transformación y carga de datos

Esta etapa abarca la selección de pre-procesamiento y transformación de datos.

Los antibiogramas son impresos por el laboratorio de análisis clínicos, los cultivos que salen positivos son enviados a la Unidad de Vigilancia Epidemiológica Hospitalaria donde son capturados en la base de datos.

Los datos de los antibiogramas negativos (donde no hubo crecimiento de microorganismos) o algunos errores de captura de datos (por ejemplo si se aislo una bacteria y se probó con un medicamento antiviral en el antibiograma) no se incluyeron.

ISBN: 978-607-97174-4-2

II-C. Minado de datos

Es una de las etapas más importantes en KDD donde se selecciona el algoritmo y la tarea que se realizará (clasificación, asociación, predicción, agrupamiento). Como resultado de este paso se obtiene un modelo de los datos que puede ser utilizado posteriormente para el análisis de los mismos.

III. Problemática

III-A. Dimensionalidad

Uno de los principales problemas al tratar de agrupar las observaciones es el número de características, debido a que cada una de estas representa una dimensión del problema y si aumentamos el número de características el espacio aumenta exponencialmente haciendo que los datos disponibles se vuelvan dispersos.

III-B. Datos mixtos

El tratamiento de bases de datos con campos mixtos (ordinales y categóricos) es un área de investigación actualmente abierta y representa un problema debido a que la mayoría de los algoritmos para agrupamiento utilizan un concepto de similitud entre dos observaciones en términos de la cercanía que existe entre estas, misma que no está definida para datos categóricos.

III-C. Interpretación

Para que los datos proporcionados sean útiles, deben de ser presentados de forma que tengan significado para el usuario objetivo, en este caso profesionales de la salud.

IV. Propuesta de solución

IV-A. Aprendizaje no supervisado

Dado que nuestro conjunto de datos no está etiquetado (no hay una columna con las salidas deseadas) es necesario seleccionar un método que nos permita ajustar un modelo de densidad a partir de las observaciones existentes.

IV-B. Agrupamiento

Es una de las tareas principales de la minería de datos exploratoria, consiste en clasificar observaciones agrupándolas por la similitud que existe entre ellas y la diferencia entre los grupos formados, para llevarla a cabo existen diversos algoritmos que varían en función del problema.

IV-C. Mapas Auto-Organizados

De acuerdo a la problemática planteada en la seccion III-A se seleccionó el algoritm mo de aprendizaje no supervisado llamado Mapa Auto-Organizado (SOM por las siglas en inglés de *Self Organizing Map*) debido a su capacidad para generar una representación discreta de baja dimensión (típicamente R^2) del espacio de las muestras de entrada de alta dimensión (R^n) [3], llamado mapa.

Esta representación se puede utilizar para visualizar correlaciones complejas entre los datos de entrada debido a que durante la etapa de entrenamiento el SOM genera un mapa topológico al acercar entre sí las observaciones que son similares mediante una función de vecindad.

El algoritmo empleado para entrenar un SOM se describe a continuación:

Algorithm 1 Algoritmo de entrenamiento de Mapa auto-organizado

Input: D, σ_0, α_0
Output: Mapa topológico con pesos adaptados
Hacer un mapa de neuronas con vectores de pesos aleatorios
for $s \leftarrow 1$ to λ **do**
  $\sigma_s \leftarrow radiusDecayFunction(\sigma_0, s, \lambda)$
  $\alpha_s \leftarrow learningRateDecayFunction(\alpha_0, s, \lambda)$
  for $t \leftarrow 1$ to $size(D)$ **do**
    $u \leftarrow bmu(neurons, D(t))$
    for $v \leftarrow 1$ to $size(neurons)$ **do**
      $d \leftarrow distance(neuron_u, neuron_v)$
      if $d \leq \sigma_t$ **then**
        $W_v(s+1) = W_v(s) + \Theta(d, \sigma_s)\alpha_s(D(t) - W_v)$
      end if
    end for
  end for
end for

Donde:

- s es la iteración actual.
- λ es la cantidad total de iteraciones.
- $D(t)$ es un vector de entrada de índice t del conjunto de datos de entrada D.
- v es el índice de una neurona en el mapa.
- u es el índice del BMU en el mapa.
- W_v es el vector de pesos de la neurona v.
- d es la distancia de la $neurona_u$ a la $neurona_v$.
- σ_s es el radio de vecindad en la iteración s.
- α_s es un restrictor de aprendizaje debido al progreso de las iteraciones.
- $\Theta(d, \sigma_s)$ es la función de vecindad.

IV-D. Mapas Auto-Organizados Generalizados

De acuerdo a la problemática planteada en la sección III-B, al utilizar datos categóricos SOM falla en preservar correctamente la topología de los datos originales, por lo que en 2006 un nuevo algoritmo GSOM [4] utiliza una nueva métrica de distancia llamada *distancia jerárquica* compuesta por *árboles de jerarquías* donde las aristas contienen un valor o *peso* y los nodos representan *conceptos* ordenados de lo más general en la raíz, hasta los más particulares en los nodos *hojas* como se muestra en la Figura 2.

Estas estructuras jerárquicas permiten manejar los distintos tipos de datos: categóricos, ordinales e incluso codificación binaria para coincidencias simples de forma unificada de allí el término de *generalización*.

IV-D1. Distancia Jerárquica: Cada punto está compuesto por una tupla *ancla* (un nodo hoja) y una ganancia (un valor real que representa la distancia desde el punto a la raíz) por ejemplo el punto $X = [\text{S. aureus}, 2]$ (ver fig. 2a).

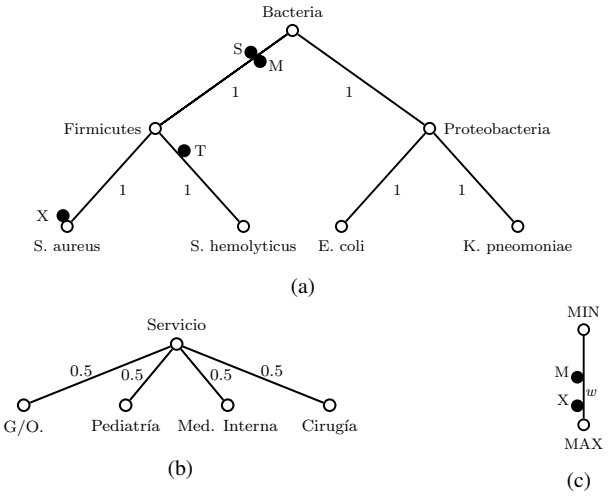

Figura 2: Tipos de distancias jerárquicas: Categórica, Simple y Ordinales.

Los patrones de entrada son mapeados a puntos en la distancia jeráruica, por ejemplo: dado el patrón de entrada (S. aureus, 7) y teniendo los árboles de jerarquías mostrados en la Fig2(a) y Fig2(c) darán como resultado un vector de puntos llamado mapa: $m = [(\text{S. aureus}, 2), (\text{MAX}, 7)]$, estos sustituyen los patrones de entrada y los pesos de las neuronas del algoritmo de SOM original.

La distancia entre dos mapas x e y es calculada utilizando (1) donde dh es un vector de distancias jeráruicas compuesto por árboles de jerarquias y el subindice i es cada componente del mapa creado del patron de entrada.

$$d(x,y) = \left(\sum_{i=1}^{n} w_i \left| dh_i(x_i) - dh_i(y_i) \right|^L \right)^{1/L} \tag{1}$$

De esta forma cada componente del patrón de entrada es mapeado y asociado a un árbol de distancia jerárquica dh_i correspondiente cuyo valor es calculado utilizando (3).

IV-D2. Distancia Jerárquica Categórica: Para poder operar con la distancia jerárquica del tipo categórico se deben de establecer algunos axiomas:

- Dos puntos en distancia jerárquica son equivalentes si se encuentran en la mísma posición aunque no tengan la misma ancla.
- Un nodo X es ancestro de otro Y si X se encuentra en el camino de Y a la raíz.
- El ancestro menos común de dos nodos X e Y denotado como $LCA(X, Y)$ por las siglas en inglés de *Least Common Ancester*.
- El punto menos común de dos nodos X e Y denotado como LCP(X, Y) por las siglas en inglés de *Least Common Point* (2) se define según sea el caso:

$$LCP(X,Y) = \begin{cases} X \text{ ó } Y, & \text{si } X \equiv Y \\ Y, & \text{si } Y \text{ es ancestro de } X \\ X, & \text{si } X \text{ es ancestro de } Y \\ LCA(X,Y), & \text{de otro modo} \end{cases} \tag{2}$$

- La distancia entre dos puntos es calculada utilizando (3):

$$|X - Y| = d_x + d_y - 2_{LCP(X,Y)} \tag{3}$$

Por ejemplo en la Fig.2a sean los puntos:

$$\begin{aligned} M &= (\text{S. aureus}, 0{,}3) \\ S &= (\text{S. hemolyticus}, 0{,}3) \\ T &= (\text{S. hemolyticus}, 1{,}3) \\ X &= (\text{S. aureus}, 2{,}0) \end{aligned}$$

- La distancia entre los puntos M y S es cero ya que son equivalentes.
- La distancia entre los puntos T y M es $(1{,}3 + 0{,}3 - 2 * 0{,}3) = 1$.
- La distancia entre los puntos X y T es $(2{,}0 + 1{,}3 - 2 * 1{,}0) = 1{,}3$ ya que el LCP(X,T) es 'Firmicutes' que tiene una distancia de 1.

Durante la fase de ajuste un punto M es ajustado a otro punto X situado en algún nodo hoja con un valor real δ y N = LCA(M, X) según el caso que corresponda:

1. Si M es ancestro de X y no se pasa de N después del ajuste entonces $d_M = d_M + \delta$
2. Si M es ancestro de X y si se pasa de N después del ajuste entonces $anchor_M = anchor_X$ y $d_M = d_M + \delta$
3. Si N es ancestro de M y no se pasa de N después del ajuste entonces $d_M = d_M - \delta$
4. Si N es ancestro de M y si se pasa de N después del ajuste entonces $anchor_M = anchor_X$ y $d_M = 2d_{N_{LCA}} - d_M + \delta$

IV-D3. Distancia Jerárquica Simple: Este tipo de distancia jerárquica opera de forma similar a homónima *categórica* pero difiere en dos aspectos: no necesita la operacion *Least Common Point* y el calculo de la distancia es 1 si tienen las misma *ancla* y 0 en caso contrario.

IV-D4. Distancia Jerárquica Numérica: En esta distancia el mapeo del patrón siempre retorna un punto donde en *ancla* siempre es *MAX* y el valor del *offset* es la distancia de X a la raíz *MIN* (ver fig. 2c) entonces la distancia entre dos puntos es calculada restando los dos valores de *offset*.

En la implementación se crean tres operaciones polimórficas para delegar la responsabilidad de mapear, calcular la distancia y ajustar los pesos a sus respectivas estructuras (ver Fig. 3), es decir que los árboles deribados de la clase *DistanceHierarcy* realizan estas operaciones de forma distinta según sea el caso.

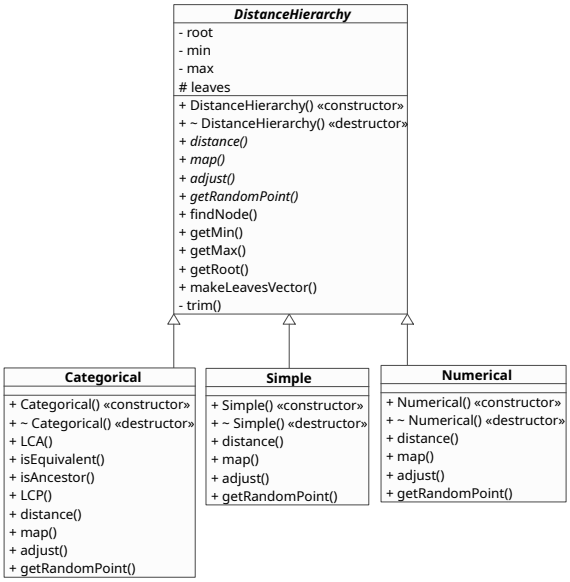

Figura 3: Jerarquía de la clase DistanceHierarcy

IV-E. Las Jerarquías en los antibiogramas

Como se vio en el apartado IV-C es posible utilizar el GSOM para visualizar los grupos cuando se tienen datos categóricos; en el caso de los antibiogramas cada patrón de entrada tiene los siguientes componentes: Fecha, Servicio, Género, Muestra, Microorganismo, Antibiótico y Susceptibilidad los cuales se intentan correlacionar, sin embargo para que estas relaciones tengan mayor significancia se crearon dos árboles de distancia categóricos sobre las variables de Microorganismos y Antibióticos utilizando en el primer caso el árbol taxonómico usado en la clasificación compuesto por el Dominio, Filo, Clase, Orden, Familia, Género, Especie (Ejemplo: Bacteria, Proteobacteria, Gammaproteobacteria, Enterobacteriales, Enterobacteriaceae, Escherichia, E. coli); en el caso de los antibióticos se utilizaron los campos de Grupo, Subgrupo y Antibiótico (Ejemplo: Aminoglucósidos, Monobactamicos, Gentamicina), se puede utilizar un árbol del tipo numérico para la fecha y el resto de los campos como una relación simple como se describió anteriormente en el apartado IV-D1.

V. Pruebas y resultados

V-A. Pruebas

Se creó un mapa auto-organizado generalizado de 10 filas por 10 columnas que se entrenó durante 100 épocas con 608 patrones de pares [microorganismo, antibótico] sin duplicados, se utilizó una taza de aprendizaje inicial de 0.9, la función de vecindad utilizada fue gaussiana con un radio inicial que abarca el mapa completo, en este caso $\sqrt{10^2 + 10^2} \approx 14{,}1421$, aplicando el redondeo hacia arriba nos queda 15.

Se crearon dos árboles jerárquicos del tipo categórico uno de microorganismos con un tamaño de profundidad de 7 y una anchura de 46 y otro árbol para los antbióticos con una profundidadad de 3 y un ancho de 48.

Las pruebas se realizaron en una Laptop DELL Inspiron 5559 con 8GB de memoria RAM con sistema operativo Arch Linux x86_64 Kernel: 5.0.0-arch1-1-ARCH

V-B. Resultados

El tiempo de entrenamiento y ejecución fue de 47.18 segundos, los resultados son mostrados utilizando una representación conocida como *Hit map* (ver fig. 4) donde los números dentro de las casillas representan el número de patrones coincidentes con la neurona en la pocición x, y y el color es asignado de acuerdo a dicho número.

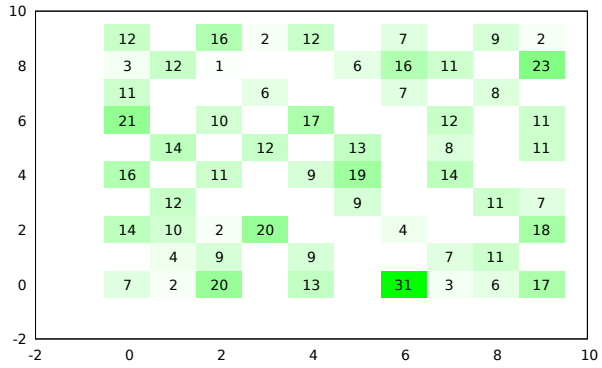

Figura 4: Mapa auto-organizado generalizado de dos distancias jerárquicas (Microorganismos y Antibióticos).

Para verificar la preservación topológica de los datos veremos en detalle el grupo formado por las neuronas $(6, 0)$, $(7, 0)$, $(8, 0)$, $(9, 0)$, $(7, 1)$ y $(8, 1)$.

- La neurona $(6, 0)$ tiene 31 coincidencias y su vector prototipo es: { Acinetobacter iwoffii, 6.00452 }, { Levofloxacina, 2.00839 }
- La neurona $(7, 0)$ tiene 3 coincidencias y su vector prototipo es: { Acinetobacter iwoffii, 6.00498 }, { Levofloxacina, 0.768032 }
- La neurona $(8, 0)$ tiene 6 coincidencias y su vector prototipo es: { Acinetobacter haemolyticus, 6.01374 }, { Ampicilina, 1.05609 }
- La neurona $(9, 0)$ tiene 17 coincidencias y su vector prototipo es: { Acinetobacter haemolyticus, 6.00209 }, { Ampicilina, 1.99333 }
- La neurona $(7, 1)$ tiene 7 coincidencias y su vector prototipo es: { Acinetobacter baumannii, 6.02939 }, { Cefotaxima, 2.00268 }
- La neurona $(8, 1)$ tiene 11 coincidencias y su vector prototipo es: { Acinetobacter baumannii, 6.00723 }, { Cefotaxima, 1.99926 }

Estas seis neuronas agrupan 75 de las 608 observaciones, todas estas corresponden a microorganismos del género Acinetobacter en contraposición por ejemplo con la neurona $(0, 6)$ en donde las 21 observaciones corresponden al género de los Enterococos.

En el caso de las neuronas $(7, 0)$ y $(8, 0)$ las observaciones coincidentes son las siguientes:

- Neurona$(7, 0)$:
  1. Acinetobacter baumannii, Moxifloxacino
  2. Acinetobacter calcoaceticus, Moxifloxacino
  3. Acinetobacter haemolyticus, Moxifloxacino
- Neurona$(8, 0)$:
  1. Acinetobacter baumannii, Imipenem y Cilastatina
  2. Acinetobacter baumannii, Meropenem
  3. Acinetobacter calcoaceticus, Imipenem y Cilastatina
  4. Acinetobacter haemolyticus, Meropenem
  5. Acinetobacter iwoffii, Imipenem y Cilastatina
  6. Acinetobacter iwoffii, Meropenem

En estos casos ambas neuronas están agrupadas de forma contigua, pero preservando las propiedades topológicas de los datos originales. Como se puede ver el primer criterio de agrupamiento es que todos pertenecen al género *Acinetobacter*, pero se encuentran en distintos grupos debido al segundo criterio de *antibióticos*, en este caso el *Moxifloxacino* pertenece al subgrupo de los *Monobactamicos* mientras que *Imipenem y Cilastatina* y *Meropenem* pertenecen al subgrupo de los *Carbapenemicos*.

VI. Conclusiones y Trabajo futuro

Los pesos de las neuronas ó vectores prototipo fueron ajustados mediante el entrenamiento utilizando una función de vecindad y una similitud de acuerdo a la distancia jerárquica.

En las pruebas realizadas la implementación preserva correctamente la topología de los datos categóricos, lo que nos permite entender la magnitud y distribución de los microorganismos según los criterios dados.

El parámetro w es un vector utilizado en (1) que puede ser utilizado para darle prioridad a un parametro dh_i, lo cuál puede conveniente al investigador para encontrar relaciones significativas en valores multidimensionales.

En el presente trabajo se utilizaron 608 patrones correspondientes a los campos categóricos de Microorganismos y Antibióticos para probar la implementación de la distancia jerárquica, sin embargo la base de datos está compuesta por 10,465 observaciones donde cada vector está compuesto por los campos de Fecha, Género, Servicio, Cultivo, Microorganismo, Antibiótico y Susceptibilidad por lo que se implementará una solución que incluya dichos campos.

Se está trabajando en una interfaz gráfica de usuario para poder variar los parámetros de entrenamiento y mostrar los resultados de forma que proporcionen mayor información y significancia.

Los mapas auto-organizados también pueden ser representados utilizando *matrices U* (matrices de distancia unificada) que son creadas utilizando la distancia promedio entre la neurona y sus vecinos más cercanos, esta distancia se puede representar en una imagen en escala de grises donde los colores claros representan nodos cercanos mientras que los colores más oscuros nodo más distantes. Por lo tanto, los grupos de colores claros se pueden considerar como agrupaciones, y las partes oscuras como los límites entre las agrupaciones. Esta representación puede ayudar a visualizar los grupos en los espacios de alta dimensión.

Una vez probado el funcionamiento de los algoritmos para la tarea dada se puede trabajar en una implementación paralela para mejorar el rendimiento.

Agradecimientos

Se agradece al Tecnológico Nacional de México, a la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de La Paz y al Consejo Nacional de Ciencia y Tecnología (CONACyT), por el apoyo recibido y las facilidades otorgadas para el desarrollo de este trabajo.

Se agradece al comité tutorial por todas sus valiosas observaciones y aportaciones para este trabajo fuera posible.

Se agradece al Benemérito Hospital General con Especialidades Juan María de Salvatierra que mediante el área de Enseñanza e Investigación a través del conducto de su titular el Dr. Gustavo Farías Noyola por haber otorgado el permiso para la utilización de una base de datos de antibiogramas.

Se agradece a la Unidad de Vigilancia Epidemiológica Hospitalaria y al Laboratorio de Análisis Clínicos del Hospital Salvatierra por haber fomentado en mí la investigación en salud y la conciencia sobre la importancia de la farmacorresistencia.

Referencias

[1] L. Gordis, *Epidemiología*, 5th ed., D. Edited by Leon Gordis, MD, MPH, Ed. Barselona, España: Elsevier, 2015.

[2] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "The kdd process for extracting useful knowledge from volumes of data," *Communications of the ACM*, vol. 39, no. 11, pp. 27–35, 1996.

[3] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, 1982. [Online]. Available: http://link.springer.com/10.1007/BF00337288

[4] C.-C. Hsu, "Generalizing self-organizing map for categorical data," *IEEE transactions on Neural Networks*, vol. 17, no. 2, pp. 294–304, 2006.