11° encuentro de investigación inaoe

4-5 de noviembre de 2010

INSTITUTO NACIONAL DE ASTROFÍSICA, ÓPTICA Y ELECTRÓNICA

XI

ENCUENTRO DE INVESTIGACIÓN

MEMORIA TÉCNICA

Tonantzintla, Puebla, México.

4 y 5 de noviembre de 2010

COMITÉ ORGANIZADOR:

Dr. Roberto Murphy Arteaga, Director de Formación Académica
Dr. Alberto Carramiñana Alonso, Coordinador de Astrofísica
Dr. Sergio Vázquez y Montiel, Coordinador de Óptica
Dr. Arturo Sarmiento Reyes, Coordinador de Electrónica
Dr. Aurelio López López, Coordinador de Cs. Computacionales
Dr. Divakara Mayya Yalia, Representante Docente de Astrofísica
Dr. Francisco Renero Carrillo, Representante Docente de Óptica
Dr. Pedro Rosales Quintero, Representante Docente de Electrónica
Dr. Ariel Carrasco Ochoa, Representante Docente de Cs. Computacionales
Martha Aurora Olmos y Flores
Gabriela López Lucio
José Luis Toxqui Olmos
Ricardo Toxqui Olmos
Fabiola Vázquez Tecuanhuey
Cecilia Tecuatl Cuautle
María Guadalupe Coyotl Tecuatl
María Esther Montes Tecanhuey
Rocío Leticia Rodas Fernández
Rosario Tlahuel Tello
Landy del Carmen Ríos Morales
Catalina Montes Tecuanhuey

EDICIÓN:

Dr. Roberto Stack Murphy Arteaga
Martha Aurora Olmos y Flores
Gabriela López Lucio
José Luis Toxqui Olmos
Ricardo Toxqui Olmos

CALLE LUIS ENRIQUE ERRO NO 1
SANTA MARÍA TONANTZINTLA, PUEBLA
C.P. 72840, TEL (222)247-27-42
http://www.inaoep.mx

Red Neuronal SOM para el Agrupamiento de Grafos Etiquetados.

Rigoberto Fonseca, Pilar Gómez-Gil, Jesús González-Bernal, Iván Olmos
Coordinación de Ciencias Computacionales, INAOE
rfonseca@inaoep.mx, pgomez@inaoep.mx, jagonzalez@inaoep.mx, iolmos@buap.mx

Resumen
Extraer conocimiento de datos estructurados es muy importante en la toma de decisiones, y una buena cantidad de estos datos utilizan grafos para representar dichas relaciones. La minería de datos basada en grafos (GDM por sus siglas en inglés) ha incursionado en los últimos años en el uso de redes neuronales como soporte para el agrupamiento, mostrando resultados iniciales prometedores. En el presente trabajo se propone una red neuronal tipo SOM capaz de agrupar grafos etiquetados no dirigidos y sin ciclos, utilizando el espectro de los grafos evaluado sobre su Laplaciano para generar prototipos de grafos de manera no supervisada. Se presentan resultados preliminares de la investigación y se discute el trabajo en proceso.

I Introducción

En la actualidad, el avance de la tecnología permite almacenar grandes volúmenes de datos, los cuales pueden tener relaciones cada vez más complejas, como son las bases de datos de proteínas, árboles filogenéticos, grafos moleculares etc. Estos datos contienen relaciones de su estructura que se pueden representar por medio de grafos. Los grafos son estructuras formadas por un conjunto de vértices y un conjunto de enlaces que son conexiones entre pares de vértices[**1**]. La extracción de conocimiento de estas bases de datos es muy importante para una correcta toma de decisiones[**2**]. Con esta finalidad nace el proceso de descubrimiento del conocimiento en bases de datos (KDD, *Knowledge Discovery in Databases*)[**3**]. El KDD busca identificar patrones válidos, nuevos, potencialmente útiles y comprensibles en datos[**3**]. Una parte indispensable de este proceso es la minería de datos. La minería de datos basada en grafos (GDM, *graph-based data mining*) es la tarea de encontrar patrones (grafos) que sean entendibles, útiles y novedosos utilizando una representación basada en grafos de los datos originales[**3**].

En la etapa de exploración de datos de KDD, las redes neuronales SOM (*Self-Organizing Maps*) han mostrado ser muy útiles[**4**]. Para realizar agrupamiento de grafos no etiquetados usando redes SOM se ha utilizado, entre otras cosas, la distancia de edición de grafos[**5**]. Calcular la distancia de edición entre el grafo A y el grafo B consiste en encontrar el mínimo número de operaciones necesarias para transformar A en B. Sin embargo, este es un proceso muy costoso. Otra propuesta[**6**], presenta una red SOM basada en representación recursiva de árboles. Por otro lado [**1**] presenta algoritmos para agrupar los vértices más conectados entre sí dentro de un mismo grafo, que muestra que usar características de los grafos (espectro) ha mostrado buenos resultados.

En este artículo se presentan los resultados preliminares obtenidos en el diseño de una extensión de la red neuronal SOM para agrupar grafos de manera no supervisada, con el objetivo de dividir el espacio de búsqueda y obtener patrones representativos de cada sub-espacio. Esta tarea es parte de un proyecto que consiste en diseñar un algoritmo de minería de datos basado en un agrupamiento no supervisado de grafos y códigos DFC[**7**], para obtener patrones útiles en la tarea de clasificación sobre datos estructurados[**8**].

El artículo está organizado de la siguiente manera: la sección II presenta algunos conceptos básicos para entender el resto del documento; la sección III presenta el modelo neuronal propuesto. Los resultados obtenidos con un pequeño experimento se muestran en la sección IV y en la sección V se resumen las conclusiones.

II Conceptos básicos

Un grafo G consiste de un conjunto de *vértices* $V(G)$ y un conjunto de *enlaces* $E(G)$, donde un enlace es un par no ordenado de vértices distintos de G; $\{u, v\}$ denota un enlace; En los grafos *dirigidos* se especifica la dirección de la

relación[9]. Un grafo pesado no dirigido G (posiblemente con lazos) tiene asociada una función de pesos $w: V \times V \to \mathbb{R}$ satisfaciendo $w(u,v) = w(v,u)$ y $w(u,v) \geq 0$. Si $\{u,v\} \notin E$, entonces $w(u,v) = 0$. La matriz de adyacencia $A(G)$ de un grafo dirigido G lo representa[10], y es la matriz entera con filas y columnas indexadas por los vértices de G, tal que la entrada uv de $A(G)$ es igual al número de arcos desde u a v (usualmente 0 ó 1). La matriz de adyacencia contiene n x n elementos, donde n es el número de vértices en el grafo.

Una posible manera de representar un grafo es a través de aplicar *análisis de componentes principales* (PCA) en su matriz de adyacencia, a fin de reducir la dimensionalidad de los datos pero retener lo más posible la variación entre los diferentes grafos. Una forma de extraer estas características de la matriz de adyacencia es por medio de sus eigenvalores. Dada una matriz A, de dimensión $n \times n$ y la ecuación:

$$AX = \lambda X \qquad (1)$$

Donde λ es un escalar y X es un vector $n \times$, si la ecuación (1) tiene una solución para λ, para un X no nulo, entonces λ se denomina *eigenvalor*, *raíz característica* o *raíz latente* de A, y el X no nulo que satisface la ecuación para un λ en particular es llamado *eigenvector*, *vector característico* o *vector latente* correspondiente para λ.

El *espectro* de un grafo se define como la lista de *eigenvalores* de su matriz de adyacencia A_G, junto con sus multiplicidades (número de apariciones). Algunos estudios mencionan que es más útil obtener el espectro a partir del Laplaciano de un grafo en vez de a partir de la matriz de adyacencia[1]; el espectro Laplaciano es el espectro de esta matriz. Dada una orientación arbitraria σ del grafo G y D la matriz de incidencia de G^{σ}, entonces el Laplaciano de G es la matriz $Q(G) = DD^T$. El Laplaciano no depende de la orientación [10]. La matriz simétrica $n \times n$ Q se denomina Laplaciano generalizado de G si $Q_{uv} < 0$ cuando u y v son vértices adyacentes de G y $Q_{uv} = 0$ cuando u y v son distintos y no adyacentes[10]. Para un grafo pesado, L se define como:

$$L(u,v) = \begin{cases} d_v - w(u,v), si\ u = v, \\ -w(u,v), si\ u\ y\ v\ son \\ \qquad adyacentes \\ 0, en\ el\ caso\ contrario \end{cases} \qquad (2)$$

Sea T la matriz diagonal formada por los grados de los vértices. El Laplaciano de G se define por

$$\mathcal{L} = T^{-1/2} L T^{-1/2} \qquad (3)$$

En otras palabras:

$$\mathcal{L}(u,v) = \begin{cases} 1 - \frac{w(u,v)}{d_v}, si\ u = v\ y\ d_v \neq 0 \\ \frac{-w(u,v)}{\sqrt{d_u d_v}},\ si\ u\ y\ v\ son \\ \qquad adyacentes \\ 0,\ en\ el\ caso\ contrario \end{cases} \qquad (4)$$

Al asignar etiquetas tanto a los vértices como a los enlaces se tiene un grafo etiquetado. En este caso se puede representar a un grafo por la 6-tupla conformada por el conjunto de vértices (V), el conjunto de enlaces (E), la lista de etiquetas de vértices (L_V) y lista de etiquetas de enlaces (L_E). Además, se requieren las *funciones de obtención de etiquetas* tanto de vértices (α) como de enlaces (β). Entonces un grafo se define como: $G = (V, E, L_V, L_E, \alpha, \beta)$ [7]. Es posible aproximar pesos a los enlaces en función de sus etiquetas y las etiquetas de sus vértices. Una manera es usando las listas de códigos VEV (Vértice-Enlace-Vértice). Asumiendo un orden lineal ($\leq_V$) en el conjunto L_V y un orden lineal ($\leq_E$) $L_{VEV} = \{c_i | i = 1, \ldots, s\}$, donde s es el número de diferentes combinaciones lineales α, β y α que existe en G (en el peor caso $s = |\alpha|^2 \cdot |\beta|$). Cada código c_i se define como la tripleta (l_a, l_e, l_b), donde, dado un enlace $e \in E$ con su etiqueta l_e, el primer componente l_a es la menor de las etiquetas de los vértices adyacentes al enlace, y l_b es la etiqueta del otro vértice[7]. Si a cada código de la lista L_{VEV} se le asigna un número entero según su posición, iniciando en 1, estos códigos funcionan como pesos en los enlaces respectivos. Entonces se puede obtener el espectro Laplaciano del grafo que aproximadamente representa al grafo etiquetado.

Los mapas auto organizados (SOM, *Self-organizing maps*)[11] son un tipo red neuronal compuesta por una capa de entrada y una o varias capas de salida. Las neuronas de entrada se conectan hacia adelante con todas las neuronas de la capa de salida (Figura 1). Las neuronas de la capa de salida, tienen cierta influencia con las que estén en su vecindario [11]. El algoritmo de entrenamiento es no supervisado. Para un número definido de iteraciones sobre el conjunto de entrenamiento, el algoritmo adapta los pesos que conectan con la neurona ganadora en un proceso de competencia entre todas las neuronas del nivel de salida. La neurona ganadora se define calculando la distancia que hay entre el vector de entrada y cada uno de los pesos de cada neurona en el nivel de salida, de acuerdo a la ecuación:

$$i(x) = argmin_i \|x - w_j\|, j = 1,2, \ldots, l \qquad (5)$$

El vecindario de influencia de la neurona ganadora se puede definir de diferentes maneras, en el presente trabajo se hace por posiciones en una rejilla, considerando los 4, 8, 12, 16, 20 y 24 vecinos más cercanos a la neurona ganadora. El coeficiente de aprendizaje que disminuye con las iteraciones se define para este trabajo como [12]:

— (6)

Donde es una constante de tiempo. La adaptación de los pesos está dada por la ecuación:

(7)

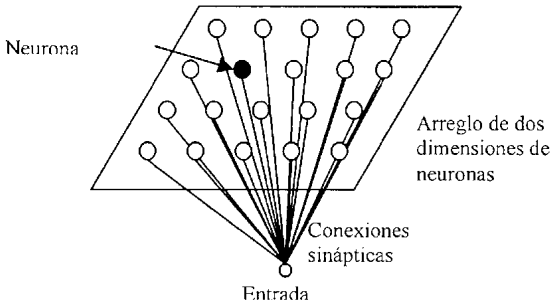

Figura 1. Ejemplo de un mapa auto-organizado[13].

III Modelo Neuronal Propuesto

El modelo se basa en extraer las características principales de cada grafo para alimentar al algoritmo SOM. Al finalizar, cada prototipo de la red SOM representará un grupo de grafos. El algoritmo propuesto es el siguiente:

1. Generar la lista de códigos VEV de todos los grafos de entrada.
2. Generar el Laplaciano de grafos pesados usando los códigos VEV.
3. Obtener el espectro de cada Laplaciano de grafo.
4. Completar cada espectro con 0's de manera que todos tengan el mismo tamaño. Debido a que los grafos pueden tener diferente numero de vértices, sus Laplacianos resultan de diferente dimensión por lo que es necesario homogeneizar ésta.
5. Entrenar una red neuronal SOM usando el espectro.
6. Agrupar cada grafo de acuerdo a la neurona ganadora al aplicarlo a la red SOM.

IV Resultados

Utilizando el algoritmo propuesto se probó con un ejemplo pequeño de once grafos. Se entrenó una red SOM con un mapa de características de 5x5 durante 500 iteraciones con un coeficiente inicial de 0.1. Los resultados se muestran en la Figura 2. La parte (a) de la figura muestra a los 11 grafos a agruparse; la parte (b) muestra la distribución que quedó de los grupos en el mapa SOM resultante después del entrenamiento; la parte (c) muestra a través de colores, los grupos en que cada grafo fue clasificado al ser evaluado por la red después del entrenamiento. Cabe aclarar que en este experimento no se utilizó un conjunto de pruebas diferente al de entrenamiento.

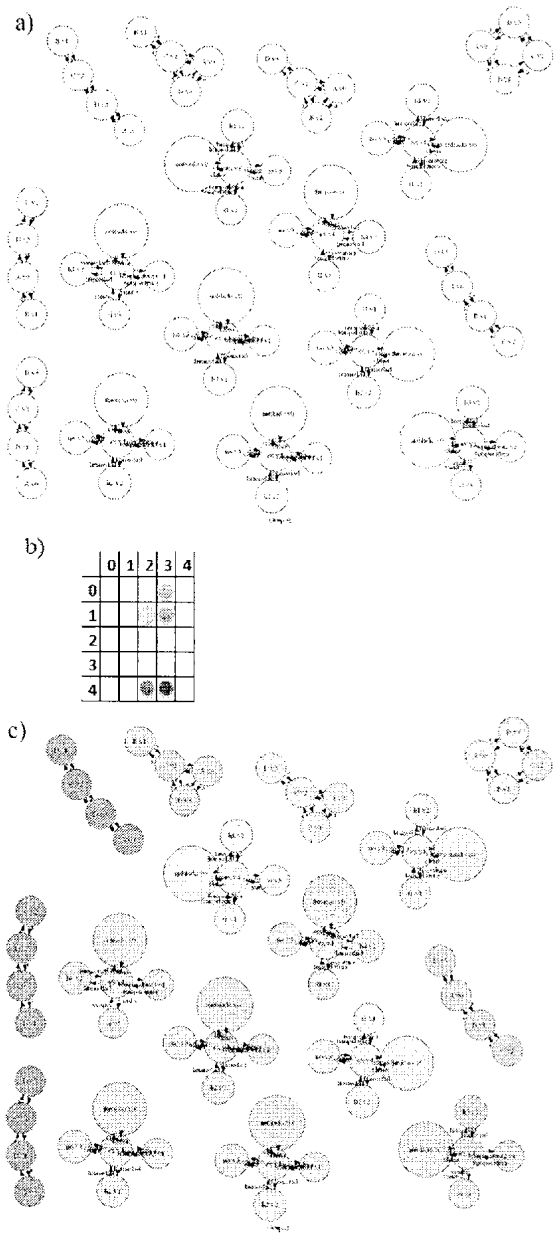

Figura 2. Ejemplo con once grafos y una red SOM de 5x5 prototipos a) grafos de entrada, b) la distribución de los grupos, c) grafos clasificados por grupo.

V Conclusiones

En este trabajo se presentan resultados preliminares obtenidos por un modelo para agrupamiento de grafos basado en sus características principales representadas a través de su espectro. Al trabajar con el espectro se consigue realizar un agrupamiento más rápido, comparado con otros modelos, ya que todos los procesos involucrados en el modelo presentado son de orden polinomial. Por otro lado, debe considerarse que se pierde información al asignar códigos a los enlaces; sin embargo esta estrategia es menos costosa que las alternativas, como sería el realizar una prueba de isomorfismo de sub-grafos etiquetados. Los resultados obtenidos a la fecha con una red neuronal SOM se mantiene la relación de los grupos, de esta forma se puede realizar traslapes entre grupos cercanos. Esto se puede aplicar en la búsqueda de patrones que sean útiles para clasificar.

VI Agradecimientos

R. Fonseca agradece al CONACyT el apoyo otorgado a través de la Beca para estudios de Maestría # 234540. Asimismo, le agradece al Lic. Pérez Galván su valioso apoyo en la elaboración del programa de cómputo desarrollado para el análisis de los datos. Este trabajo fue parcialmente apoyado por el Proyecto CONACYT 88990-B.

VII Referencias

[1] Schaeffer, Satu Elisa. Graph clustering. *Computer Science Review*, 1, 1 (2007), 27-64.

[2] Da San Martino, Giovanni and Sperduti, Alessandro. Mining Structured Data. *IEEE Computational Intelligence Magazine*, 5, 1 (2010), 42-49.

[3] Fayyad, Usama, Piatetsky-Shapiro, Gregory, and Smyth, Padhraic. From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, 17, 3 (1996), 37-54.

[4] Vesanto, Juha and Alhoniemi, Esa. Clustering of the Self-Organizing Map. *IEEE Transactions on Neural Networks*, 11, 3 (2000), 586-600.

[5] Günter, Simon and Bunke, Horst. Self-organizing map for clustering in the graph domain. *Pattern Recognition Letters*, 23, 4 (2002), 405–417.

[6] Hagenbuchner, Markus, Tsoi, Ah Chung, Sperduti, Alessandro, and Kc, Milly. Efficient Clustering of Structured Documents Using Graph Self-Organizing Maps. In *Focused Access to XML Documents*. Springer, Berlin / Heidelberg, 2008.

[7] Olmos Pineda, Ivan. *Common Subgraph Search Based on Vertex-Edge-Vertex Codes and a Breadth-Depth Search, tesis sometida como requisito para obtener el grado de Doctor en Ciencias Computacionales*. INAOE, Tonantzintla, 2006.

[8] Fonseca Delgado, Rigoberto Salomón, González Bernal, Jesís Antonio, Gómez Gil, María del Pilar, and Olmos, Iván. *Propuesta de tésis: Diseño de un algoritmo de minería de datos basada en grafos para la tarea de aprendizaje de conceptos*. Instituto Nacional de Astrofísica Óptica y Electrónica, Tonantzintla, 2010.

[9] Vicens Salort, Eduardo, Órtiz Bas, Ángel, and Guarch Bertolín, Juan José. *Métodos Cuantitativos*. Servicio de Publicaciones, Valencia, 1997.

[10] Godsil, Chris and Royle, Gordon. *Algebraic Graph Theory*. Springer-Verlang, New York, 2001.

[11] Kohonen, Teuvo. The self-organizing map. *Proceedings of the IEEE*, 78, 9 (1990), 1464-1480.

[12] Haykin, Simon. *Neural Networks*. Pearson Prentice Hall, Delhi, India, 2005.

[13] Gómez Gil, María del Pilar. In *Redes Neuronales Avanzadas* (Julio 2010), http://ccc.inaoep.mx/~pgomez.