Problema CSP y DisCSP

Ramón Trinidad Acedo

2026-01-02

Índice de contenidos

1. Introducción

Este trabajo se enmarca en el contenido de la asignatura TSCD. Intentaremos afianzar los contenidos aprendidos sobre el lenguaje de la representación para grafos y las Graph Neural Networks.

A pesar de los prometedores avances del Deep Learning aplicado en datos tabulares y de la gran eficiencia de los modelos aplicados para la clasificación y la regresión, en ocasiones este tipo de representacion de los datos presenta una débil modelización de las relaciones latentes entre instancias y carácteristicas. Como solución a este problema surge la idea de recoger estas interaciones en los datos mediante la aplicación de Graph Neural Networks.

El objetivo de este proyecto será implementar Graph Neural Networks for Tabular Data Learning (GNN4TDL). En primer lugar, descargaremos una base de datos tabular, en este caso será...., estableceremos ciertos criterios con los que construir un grafo a partir de estos datos. Una vez aquí, ya podemos aplicar técnicas como Node2vec para el embedding de los nodos. Por último plantearemos una GNN que sea capaz de leer las relaciones de nuestro grafo con embeddings eficientes y que resuelva el problema de clasificación.

2. ¿Por qué es útil el uso de GNNs para TDL?

Las razones por las que las GNNs pueden beneficiar el aprendizaje con datos tabulares son las siguientes:

- **Correlación entre instancias:** Usualmente, consideramos que los datos recogidos son realizaciones de una muestra aleatoria simple, y que por lo tanto las instancias son independientes. No obstante, instancias con carácteristicas similares suelen ser clasificadas con la misma etiqueta. Por ejemplo, pacientes con datos clínicos similares suelen tener probabilidades similares de tener una cierta enfermedad.

  Con una correcta formulación de la estructura de la red, una GNN puede captar estas relaciones entre instancias.

- **Interación entre características:** Se refiere al efecto de combinar carácteristicas. Convencionalmente estas interacciones se modelan a mano, sin embargo, considerando carácteristicas como nodos en un grafo, las aristas pueden representar las potenciales interacciones, por lo que las GNN pueden aprender estás combinaciones de manera más sofisticada.

- **Relación de alto orden:** En los modelos convencionales asumimos que dos elementos con carácteristicas similares obtendrán una predicción similar. Esta idea se basa en la directa interacción entre instancias y carácteristicas. Sin embargo, GNNs pueden modelar interacciones de alto orden entre los elementos de los datos de manera eficiente. Al transformar nuestros datos tabulares en una red compleja estamos desentrañando una estructura implícita en los datos que antes no podíamos ver y que permite a las GNNs aprender mediante mensajes y agregaciones de nodos vecinos y nodos vecinos de vecinos, etc.

- **Semi-supervisión:** En casos donde recopilar datos tabulares etiquetados suele ser complicado, una de las carácteristicas principales de las GNNs es el aprendizaje semi-supervisado, el cual propaga la información supervisada (con etiquetas) a las no etiquetadas gracias a la estructura del grafo.

  Además, avances recientes han tratado de desarrollar tareas auto-supervisadas en el proceso de aprendizaje de representación de instancias tabulares. De manera que, podemos mejorar la red resultante de datos a la vez que entrenamos la GNN para su tarea principal de clasificación o regresión.

- **Capacidad inductiva:** Las GNNs exhiben la habilidad y el poder inductivo de generalizar el mecanismo de transmisión de mensajes a nodos nunca vistos e incluso a nuevas redes. Cuando instancias y carácteristicas son representadas como nodos y sus interacciones como aristas en una red, aprender las representaciones con GNNs permite trabajar con características no vistas directamente en el set de prueba, producir embeddings para nuevas instancias y generalizar sobre tareas nunca aprendidas en el entrenamiento.

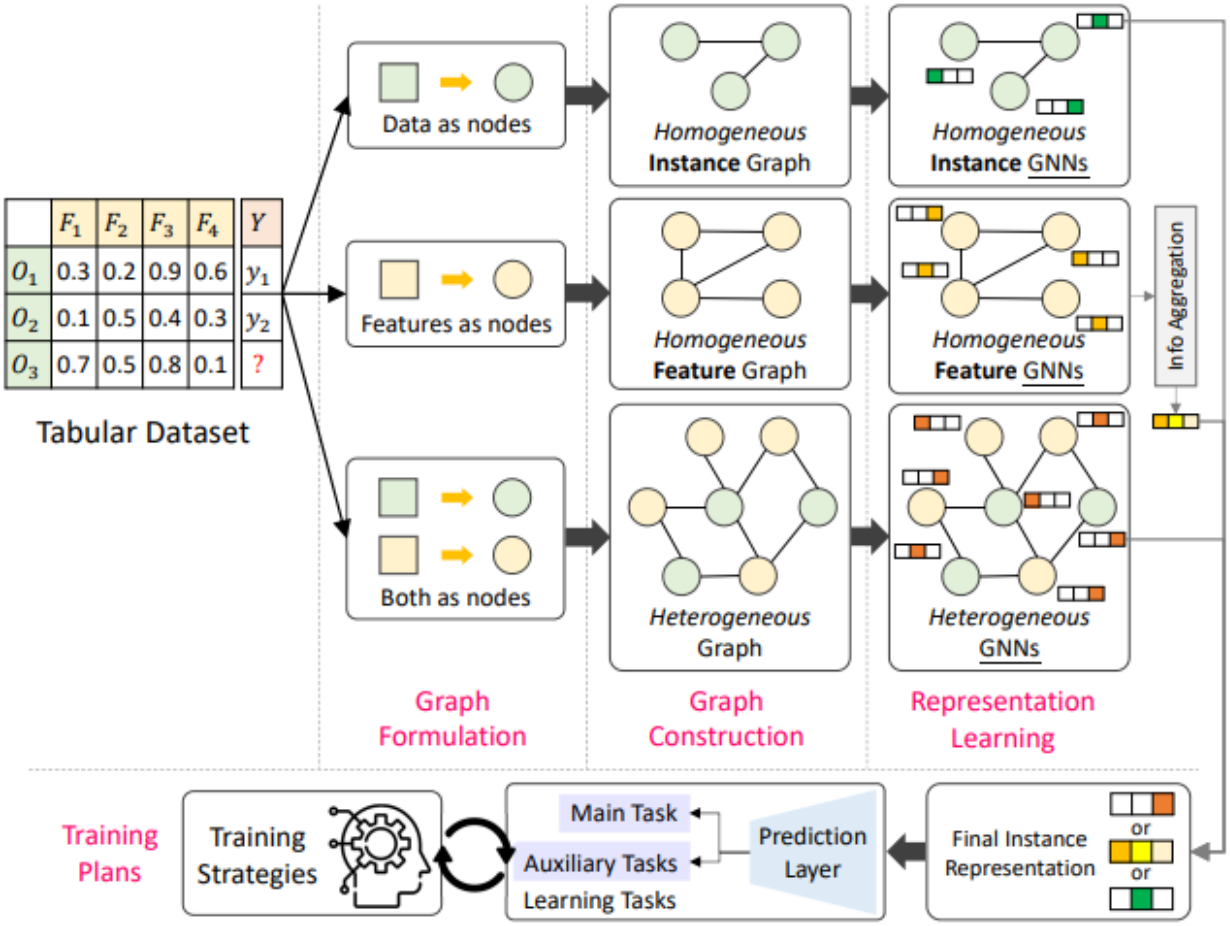

Figure 1: Pipeline general de la metodología GNN4TBL.

3. Pipeline para GNN4TBL

En esta sección vamos a desarrollar un plan de trabajo para GNN4TBL, estableceremos los pasos necesarios para implementarlo e indagaremos en la cantidad de posibilidades que nos ofrece esta metodología.

3.1. Formulación y construcción de la red

Naturalmente, el primer paso que debemos dar es establecer los criterios para la construcción del grafo a partir de los datos tabulares. Esta fase incluye las siguientes acciones:

3.1..1 Elección de los nodos

Debemos elegir que elementos usaremos como nodos.

1. **Grafos de instancias:** Dan lugar a grafos homogéneos creados para modelar las relaciones globales entre las muestras de datos, por lo que su construcción depende de todas las carácteristicas. De hecho, a este tipo de grafos les resulta díficil capturar relaciones localmente en términos de subconjuntos de características.

   Además, el uso de este tipo de grafos suele restringirse a cuando el número de características es pequeño, por ejemplo en SUBLIME (Liu et al. 2022), LUNAR [44], SLAPS [33] e IDGL con menos de 65 variables. Esto se debe a que es mucho más simple e interpretable a la hora de relacionar los nodos, y para evitar el ruido, pues conjuntos de datos con muchas carácteristicas y no todas significativas o beneficiosas puede llevar a obstaculizar el rendimiento de la red.

2. **Grafos de características:** Consiste en modelar las interacciones entre características en un grafo homogéneo, en el cual los nodos son las caracteristicas y las aristas sus correlaciones. Uno de los objetivos de este tipo de redes es aprender las interacciones de alto orden entre carácteristicas e instancias que antes comentabamos. Es bueno resaltar que aunque las relaciones entre características se aprenden de manera automática, en ciertos campos como la medicina se requiere tener un conocimiento de dominio significativo para contrastar las dependencias de algunas características

3. **Grafos heterogéneos:** Combinación de instancias y carácteristicas como nodos. Pueden aprender relaciones más complejas y diversas, tanto instancia- instancia, caracteristica-característica o instancia-característica dando de esta manera una más rica y comprensiva representación de los datos tabulares. Considerando que nodos y aristas pueden representar diferentes elementos e informaciones, podemos clasificar este tipo de grafo en varias categorias.

   - **Generales:** Se toman los valores categóricos de las características como nodos y conectarlos a los nodos de las instancias correspondientes.
   - **Bipartitos:** Los nodos se agrupan en dos tipos diferentes y solo se pueden conectar con aristas nodos de distinto tipo. Naturalmente, se suele considerar un tipo de nodo para las intancias y el otro para las características (o para sus valores) y los valores de las caracteristicas como pesos de las aristas.

     Algunas de las ventajas de los grafos bipartitos es que mantienen la estructura original de los datos tabulares, soportan todo tipo de variable (categóricas, numéricas, etc.) simplemente añadiendo propiedades a las aristas, solucionan el problema de los valores faltantes simplemente no uniendo los correpondientes nodos, y representan una forma eficiente de calcular la proximidad entre instancias.
   - **Grafos multirelacionales:** Mucho más complejos, pueden contener varios tipos de nodos y aristas.
   - **Grafos de conocimiento:** Ofrecen información auxiliar para modelar las relaciones entre las caracterñisticas de un conjunto de datos tabular. Se suelen usar recursos externos y conocimientos del dominio para construirlos.

3.1..2 Construcción del grafo

Seleccionados el tipo de nodos, podemos aplicar varias metodologías para crear las aristas del grafo y elegir la correcta es fundamental para la calidad de la estructura del grafo. Demasiadas aristas pueden hacer que la red sea demasiado abstracta y computacionalemente compleja, mientras que pocas aristas derivan en un grafo demasiado desconectado y con poca información.

- **Estructura intrínseca:** Consiste en respetar la estructura intrínseca del dataset tabular (filas y columnas).

Es un método muy intuitivo y común, sobretodo en la formulación de grafos bipartitos, donde ya hemos visto que conservan las relaciones iniciales de los datos. Si una instancia posee una específica categoría en una cierta característica, creamos una arista entre ellas.

No obstante, aunque la estructura intrínseca de los datos tabulares otorgan una base para la construcción del grafo, el potencial que nos ofrecen las GNNs nos lleva en muchos casos a complementar este método con otras técnicas.

- **Métodos basados en reglas:** Se siguen criterios predefinidos o heuristícas para determinar los nodos y las aristas del grafo resultante. Normalmente, se usan instancias y carácteristicas como nodos y las aristas se rigen mediante condiciones lógicas y reglas.

  Este método suele ser aplicado, por ejemplo, en grafos de instancias, donde dos nodos serán conectados si comparten una cierta categoría o si la similaridad definida entre ellos supera un cierto umbral.

  Además, las reglas pueden estar basadas en conocimiento, correlaciones u objetivos específicos. Los criterios más conocidos son **k-vecinos más cercanos** (unir los nodos más cercanos según una determinada similaridad), unir los nodos cuya similaridad supere un cierto **umbral**, unir **todos con todos** y unir los que **compartan valores** de algún atributo.

- **Métodos de aprendizaje:** Se entrenan modelos que buscan determinar automáticamente la estructura óptima de un grafo capturando información y relaciones complejas de manera eficiente. Se pueden categorizar en cuatro grupos:

  - **Basados en métricas:** Usan funciones kernel para computar la similaridad entre las propiedades de los nodos y usan esta similaridad como peso de las aristas. Una ventaja importante es que ofrecen la posibilidad de entrenar el modelo debido a que los kernel son diferenciables.
  - **Enfoque neuronal:** emplea redes neuronales sofisticadas para determinar el peso de las aristas.
  - **Enfoque directo:** Se considera la matriz de adyaciencia del grafo como un conjunto de parametros que deberemos estimar. A diferencia de los otros métodos no depende de las propiedades de los nodos para definir las aristas pero es costoso computacionalmente.

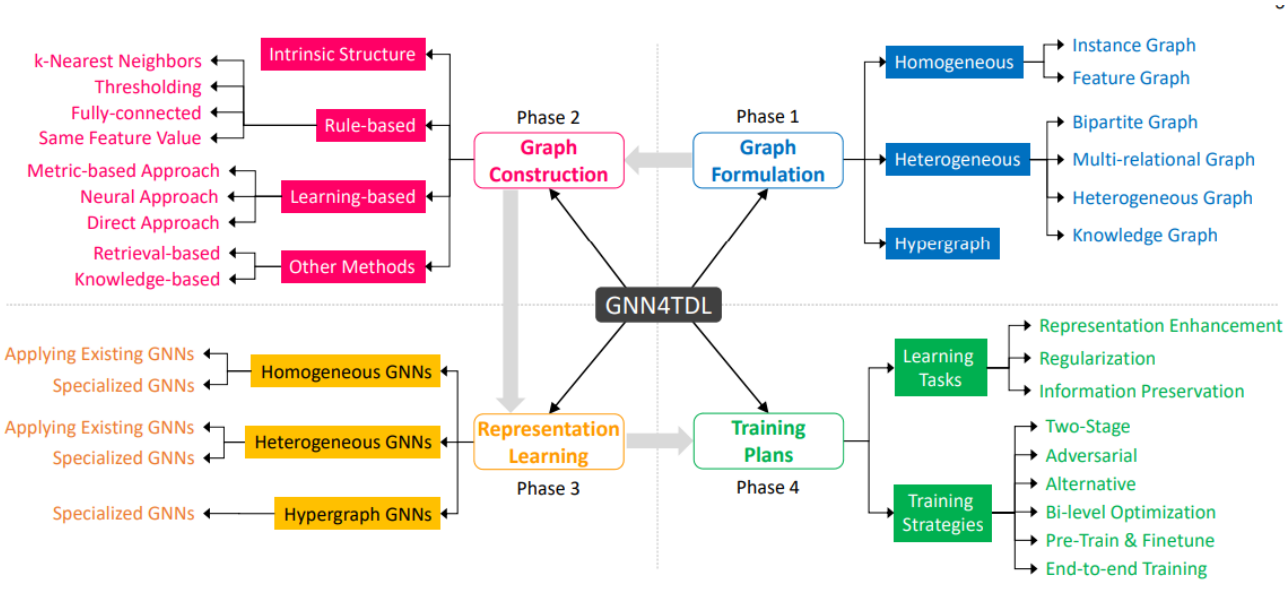

Figure 2: Resumen de los criterios de formulación de la red.

4. Aprendizaje de la representación

5. Plan de entrenamiento

> caja

Referencia de ejemplo: Knuth discute algoritmos clásicos (Knuth 1968).

Bibliografía

Knuth, Donald E. 1968. *The Art of Computer Programming, Vol. 1.* Addison-Wesley.

Liu, Yixin, Yu Zheng, Zhang Daoukun, Hongxu Chen, Hao Peng, and Shirui Pan. 2022. "Towards Unsupervised Deep Graph Structure Learning." *Proccedings of the ACM Web Conference 2022*, 1329–1403.