

Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional Unidad Guadalajara

Razonamiento causal en selección de algoritmos de toma de decisiones

COMPUTACIÓN II - ALGORITMOS PARA LA TOMA DE DECISIONES

Alumna:

Liliana Durán Polanco

Profesor:

Dr. Mario Angel Siller
Gonzalez Pico

Abstract: La Taxonomía para Toma de Decisiones bajo incertidumbre en sistemas IoT (TTDIoT) ofrece una caracterización de los elementos que intervienen en la selección de un algoritmo para la toma de decisiones, su objetivo es servir como una guía de diseño, sin embargo, TTDIoT aun depende fuertemente de un humano para ser aplicada. En este trabajo se explora la posibilidad de representar el problema de elegir una técnica algorítmica como una estructura Bayesiana, donde las categorías de TTDIoT representen variables aleatorias con relaciones de causalidad. Se exploraron tres algoritmos de aprendizaje de estructura y se realizó una prueba de inferencia con las estructuras obtenidas. Los resultados muestran que una estructura Bayesiana es un buen modelo para este problema, sin embargo, se requiere un conjunto de datos mayor para encontrar las distribuciones conjuntas adecuadas.

Key-words: Bayesian network, TTDIoT, Aprendizaje de estructura

1. Introducción

La Taxonomía para la Toma de Decisiones bajo incertidumbre en sistemas IoT (TTDIoT) [8] contiene un conjunto de características asociadas a la solución algorítmica de un problema de toma de decisiones en el dominio del IoT, su objetivo es establecer relaciones entre los diferentes elementos que componen un problema y su solución. El proceso de crear un algoritmo de decisión dado un conjunto de requerimientos es un proceso de ingeniería. Newell y Simon [7] abstraieron el proceso de ingeniería como un proceso de resolución de problemas; este proceso es descrito como un proceso de búsqueda a través de un espacio de estados, donde los estados representan las soluciones. Si el problema de seleccionar un buen algoritmo de toma de decisiones es definido por medio de una especificación de requerimientos es posible encontrar un algoritmo que se ajuste. Desde un punto de vista computacional, la resolución de problemas requiere la representación de conocimientos. Esta representación es fundamental para explorar eficientemente el espacio de búsqueda [4].

Una red Bayesiana es un modelo causal. Un modelo causal representa los principios o características del dominio que se modela, y las asociaciones entre estos elementos representan causas y efectos [1]. Una red Bayesiana es un modelo que puede representar el conocimiento sobre un problema y dados los supuestos probabilísticos bajo los que esta construida permite inferir soluciones. Dado que TTDIoT contiene un conjunto de características que tienen un efecto sobre la elección de un algoritmo de toma de decisiones una red Bayesiana representa un modelo adecuado que permite realizar inferencias, pero también determinar como influyen las diferentes características entre sí.

El objetivo de este trabajo es explorar una red Bayesiana como modelo para inferir soluciones a problemas de decisión en el dominio del IoT dada una especificación de requerimientos basada en TTDIoT. En este trabajo se utilizan tres algoritmos de aprendizaje de estructuras Bayesianas usando los datos obtenidos de la revisión sistemática desarrollada en el trabajo [8], y posteriormente se evalúan utilizando una combinatoria de las categorías en TTDIoT comparando, si es posible, con la clasificación de la revisión sistemática.

La sección 2 presenta el marco teórico del trabajo, la sección 3 presenta la metodología utilizada durante el presente estudio, en la sección 4 se muestran y analizan los resultados obtenidos, y finalmente, en la sección 5 se exponen las conclusiones.

2. Marco teórico

2.1. Representación de un problema

En este trabajo tomamos como base la definición de [5], un problema es representado por estructuras de datos y programas que operan sobre ellas para hacer inferencias.

2.1 Red Bayesiana y razonamiento causal

Una red Bayesiana es un grafo aciclico dirigido (DAG) que modela independencia y relaciones causales. Cada nodo X_i corresponde a una variable aleatoria, y la distribución de la probabilidad conjunta $p(X_1, \dots, X_N)$ es dada por las distribuciones condicionales de cada nodo X_i dados sus padres $pa(X_i)$:

$$p(X_1 : N) = \prod_{i=1}^N p(X_i | pa(X_i))$$

Las aristas en una red Bayesiana representan una relación semántica. Sí existe un camino dirigido de X_i a X_j entonces X_i es una causa de X_j . Este efecto causal es la distribución condicional de X_j dado X_i [3].

2.2. Aprendizaje de estructura

La estructura de una Bayesiana (DAG) puede ser aprendida dado un conjunto de datos. El aprendizaje consiste en obtener la topología de la red o el DAG y la distribución de probabilidad para cada variable aleatoria, sin embargo, este es un proceso complejo debido a que el número de posibles estructuras es de orden exponencial y requiere una gran cantidad de datos. Muchos métodos han sido desarrollados para resolver este problema existiendo dos enfoques principales: métodos globales y métodos locales [10].

Los *métodos globales* utilizan una heurística para navegar el espacio de búsqueda. Inician con una estructura inicial, se utiliza una métrica para evaluar que tan bien la estructura modela los datos, y se generan variaciones a cada paso.

Los *métodos locales* evalúan las relaciones de independencia entre subconjuntos de variables para obtener la estructura, sin embargo, requieren una gran cantidad de datos.

2.2.1 Bayesian Score

Bayesian score (BD) es una métrica para evaluar lo bien que un modelo se ajusta a los datos. Esta métrica esta basada en el teorema de Bayes, y consiste en obtener la probabilidad posterior de una estructura (G_i) dados los datos (D):

$$P(G_i | D) = \frac{P(G_i)P(D|G_i)}{P(D_i)}$$

Esta métrica se basa en cuatro supuestos:

- Muestra multinomial (solo puede aplicarse para variables discretas).
- Se asume que la estructura sigue una distribución previa Dirichlet.
- Los parámetros (θ) son independientes.
- Modularidad de parámetros (θ). Dados dos DAG G y G' , tal que $P(G) > 0$ y $P(G') > 0$, entonces: $p(\theta|G) = p(\theta|G')$. Los parámetros representan las distribuciones conjuntas pertenecientes a las variables aleatorias.

2.2.2 Algoritmo Hill-Climbing

Una estrategia común para encontrar la mejor estructura dada una métrica como BD, es utilizar la estrategia de búsqueda Hill-Climbing que consiste en evaluar una estructura inicial que es mejorada a cada paso. Un algoritmo que utiliza esta estrategia trabaja así:

1. Genera un DAG inicial.
2. Calcula la métrica de evaluación.

3. Se añade/invierte/remueve una arista del DAG.
4. Se calcula la métrica de evaluación para la nueva estructura.
5. Si la métrica mejora, se mantiene el cambio. Si no mejora, se regresa al DAG previo.
6. Los pasos 3-5 se repiten hasta que no haya mejoras.

La estrategia Hill-Climbing no garantiza encontrar una estructura óptima porque puede quedarse en valores óptimos locales.

2.2.3 Algoritmo K2

El algoritmo K2 utiliza un ordenamiento de las variables para reducir el número de estructuras a evaluar. A este ordenamiento se le conoce como orden causal. Dado un ordenamiento causal, aprender la mejor estructura es equivalente a seleccionar el mejor conjunto de padres para cada nodo en la red. Dado el orden, el algoritmo K2 se encarga de agregar padres a cada nodo para incrementar el puntaje de evaluación que puede ser el Bayesian Score (BD). Cuando no se incrementa el puntaje la búsqueda se detiene.

3. Metodología

Con el fin de explorar una red Bayesiana como modelo para inferir algoritmos de toma de decisiones dada una especificación de requerimientos basada en TTDIoT, se ha decidido utilizar tres diferentes algoritmos de aprendizaje, la razón es que los algoritmos de aprendizaje pueden encontrar una estructura óptima y esta puede ser diferente según el algoritmo implementado. Para este trabajo se han seleccionado solo algoritmos que pertenecen al grupo de *métodos globales* porque el conjunto de datos basado en TTDIoT es pequeño. La metodología utilizada es mostrada en la Fig. 1 y los pasos se detallan a continuación.

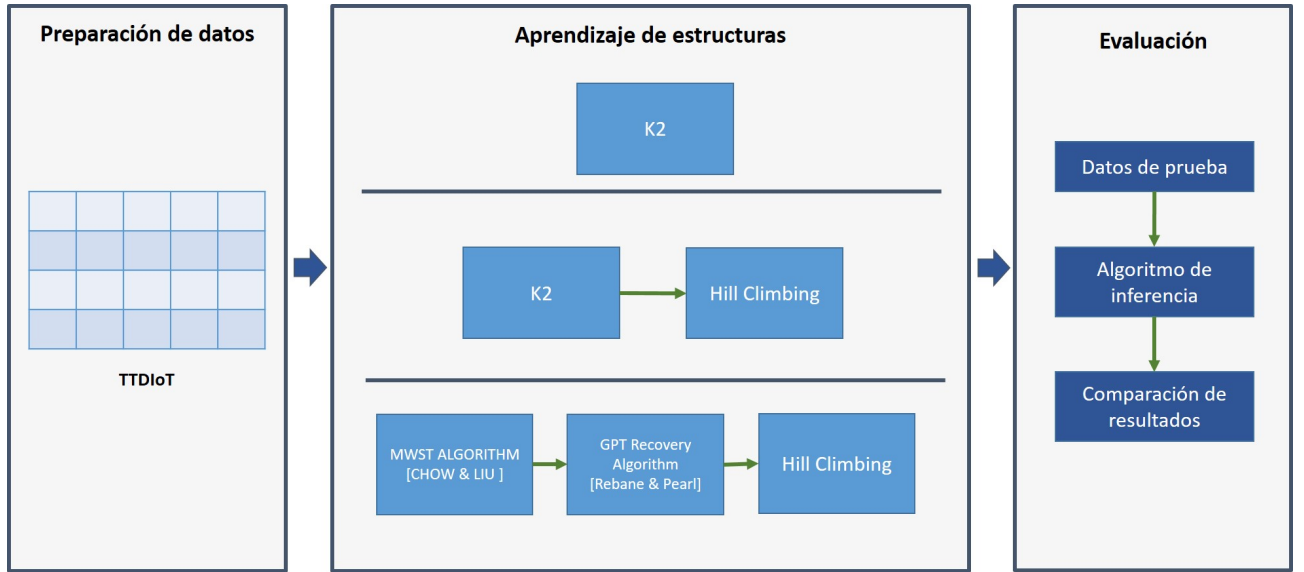


Figure 1: Metodología

3.1. Preparación del conjunto de datos

La clasificación resultado de la revisión sistemática desarrollada en [8] es utilizada como conjunto de datos para aprender la estructura. El conjunto de datos esta formado por 57 elementos u observaciones del proceso de selección de un algoritmo de toma de decisiones. Cada observación esta formada por seis categorías: *Objetivo*, *Procesamiento*, *Colaboración*, *Incertidumbre*, *Enfoque algorítmico (Nivel 1)*, *Enfoque algorítmico (Nivel 2)*. Cada categoría toma valores nominales. Con el fin de manejar fácilmente el conjunto de datos se asocian valores enteros a cada posible valor dentro de una categoría, ver Tab. 3.1, 3.2.

3.2. Aprendizaje de estructura

Tres algoritmos fueron seleccionados para aprender la estructura: *K2*, *K2+Hill Climbing*, *MWST+GPT recovery+Hill Climbing*. Estos algoritmos utilizan como metrica de evaluación el BD. A continuación se presentan detalles sobre la implementación de estos algoritmos.

Objetivo			
1 Predicción			
2 Optimización			
3 Planeación			
4 Pronostico			
5 Resolución de conflicto			
(a) Categoría 1			

Procesamiento			
1 Centralizado			
2 Descentralizado			
(b) Categoría 2			

Colaboración			
1 Humano			
2 Sistema			
(c) Categoría 3			

Incertidumbre			
1 Medida			
2 Parámetro			
3 Modelo			
(d) Categoría 4			

Table 3.1: Categorías TTDIoT

Enfoque algorítmico (Nivel I)			
1 Probabilístico			
2 Teoría de utilidad			
3 Basado en datos			
4 Basado en criterio			
(a) Categoría 5			

Enfoque algorítmico (Nivel II)			
1 Bayesiano			
2 Restricción			
3 Red de decisión			
4 Programación dinámica			
5 Teoría de la evidencia			
6 Teoría de juegos			
7 Heurística			
8 Multicriterio			
9 Estocástico			
10 Aprendizaje supervisado			
11 Aprendizaje no supervisado			
(b) Categoría 6			

Table 3.2: Categorías

3.2.1 K2

El algoritmo K2 supone un orden causal sobre las variables, en este caso, el orden es determinado por semántica externa. La implementación utilizada para este algoritmo fue tomada del libro [6]. El algoritmo toma como entrada el conjunto de datos D , un arreglo con las variables *vars* y el orden causal de las variables *ordering*. El orden dado es: **Objetivo, Procesamiento, Colaboración, Incertidumbre, Enfoque Algorítmico Nivel 1, Enfoque Algorítmico Nivel 2**. El algoritmo K2 retorna un grafo que representa la estructura con la mejor puntuación de acuerdo al BD.

3.2.2 K2+Hill Climbing

El algoritmo K2 supone que la mejor estructura es determinada por la mejor selección de padres, sin embargo, es posible que exista una estructura con un mejor puntaje por ello se ha decidido utilizar, también, el algoritmo Hill Climbing. La implementación del algoritmo Hill Climbing es tomada del libro [6].

El algoritmo K2 retorna un grafo que representa la estructura con la mejor puntuación de acuerdo al BD; este grafo se toma como el grafo inicial para el algoritmo Hill Climbing, que también retorna un grafo. El grafo resultado de esta secuencia representa una mejora en cuanto al BD.

El orden dado es: **Objetivo, Procesamiento, Colaboración, Incertidumbre, Enfoque Algorítmico Nivel 1, Enfoque Algorítmico Nivel 2**. Hill Climbing requiere como parámetro un grafo inicial, que en este caso se toma de la salida del algoritmo K2.

3.2.3 MWST+GPT recovery+Hill Climbing

El algoritmo Hill Climbing implementado en el libro [6] requiere un grafo inicial y aleatoriamente cambia las aristas buscando un mejor grafo. Esta implementación tiene como desventaja dejar nodos desconectados, además si se parte de un mal grafo inicial encontrar el grafo óptimo se vuelve tardado. Para eliminar estas desventajas se decidió crear un grafo previo a la implementación del algoritmo Hill Climbing para reducir el espacio de búsqueda.

Creando un grafo inicial

1. El algoritmo MWST (Maximum Weighted Spanning Tree) desarrollado por [2], permite crear un árbol basándose en la métrica de información mutua I . La métrica de información mutua I determina la influencia de una variables sobre otra.

$$I(X_i, X_j) = \sum_{X_i, X_j} P(X_i, X_j) \log \frac{P(X_i, X_j)}{P(X_i)P(X_j)}$$

El algoritmo calcula la métrica de información mutua para cada par de variables y obtiene el *Maximum Weighted Spanning Tree*. Obtener el *Maximum Weighted Spanning Tree* consiste en agregar codiciosamente aristas con el mayor peso evitando ciclos. El peso esta representado por la métrica de información mutua para cada par de variables. La salida de este algoritmo es un árbol (Grafo sin ciclos) con aristas no dirigidas. La implementación de este algoritmo se puede encontrar en el apéndice A.

2. El algoritmo GPT (Generating Poly-tree) Recovery [9] toma como entrada el árbol generado por el algoritmo MWST y tiene por objetivo encontrar los nodos independientes para entregar como salida un grafo semi-dirigido. GPT Recovery busca en el árbol estructuras de la forma $X - Y - Z$ y realiza pruebas de independencia estadística para determinar si dos variables son independientes, si X y Z resultan se independientes sus aristas de direccionan hacia Y . Este proceso se repite hasta que se han procesado todas las estructuras de la forma $X - Y - Z$. Para la implementación de este algoritmo se utilizo el test de independencia χ^2 con $\alpha = 0.05$. La implementación de este algoritmo se puede encontrar en el apéndice A.

Usando Hill Climbing

1. El grafo semidirigido obtenido del algoritmo GPT Recovery, representa un grafo de Markov equivalente. Hill Climbing debe encargarse de buscar la dirección de aquellas aristas aun no dirigidas. Por lo tanto, la implementación presentada en el libro [6] tuvo que ser modificada, esta modificación puede ser encontrada en el apéndice A.

3.3. Evaluación

La evaluación de las estructuras obtenidas en el paso anterior se hará sobre un conjunto de especificaciones de acuerdo a TTDIoT. Este conjunto se crea con la combinatoria de las categorías: *Objetivo*, *Procesamiento*, *Colaboración e Incertidumbre*. El objetivo de la prueba es obtener el *Enfoque algorítmico (Nivel II)* inferido por la estructura dado un vector de cuatro categorías.

El conjunto de datos de prueba contiene 60 elementos, cada elemento se pasará como parámetro a un algoritmo de inferencia por eliminación de variables. Este algoritmo recibe un ordenamiento de las variables aleatorias en la red Bayesiana (Ordenamiento: **Enfoque Algorítmico (Nivel II)**, **Enfoque Algorítmico (Nivel I)**, **Incertidumbre**, **Colaboración**, **Procesamiento**, **Objetivo**), y se genera un conjunto de datos con el Enfoque Algoritmo (Nivel II) inferido para análisis. La Fig. 2 ilustra este proceso.

Se cuenta con una pequeña clasificación de algoritmos dada por la revisión sistemática desarrollada en [8]. Sin embargo, la mayoría de los valores dentro de la combinatoria no fueron abordados en los trabajos clasificados, por ello, solo cuando sea posible se comparará el resultado con esta clasificación.

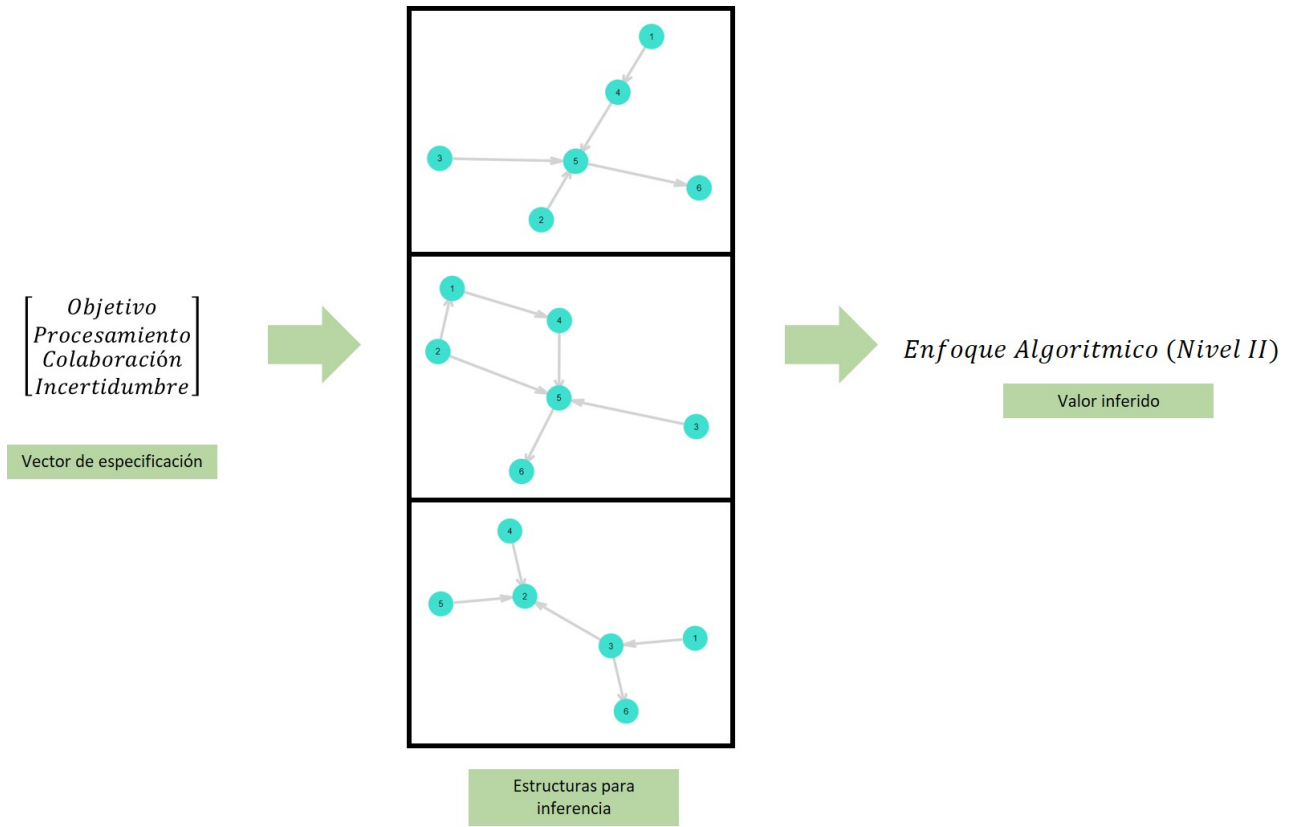


Figure 2: Proceso de prueba

4. Resultados

4.1. Resultados con estructura obtenida con K2

4.1.1 DAG

El grafo obtenido de la ejecución del algoritmo K2 (Fig. 3), presenta una relación de causalidad hacia el *Enfoque Algoritmico (Nivel II)*. Se pueden observar relaciones de causalidad interesantes, por ejemplo, el *Objetivo* tiene un efecto directo sobre la *Incertidumbre*, mientras que *Colaboración* y *Procesamiento* lo tienen sobre el *Enfoque algoritmico (Nivel I)*. Esta estructura obtuvo como evaluación final $BD_{final} = -318.81$.

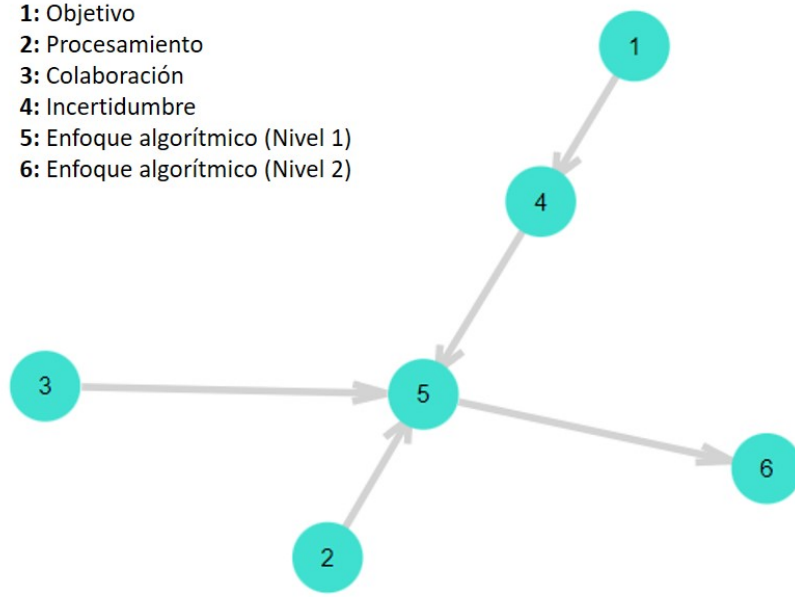


Figure 3: DAG obtenido con K2

4.1.2 Resultados inferencia

En la Fig. 4 se muestra la distribución de resultados inferidos. La estructura pudo inferir solo cuatro de las once técnicas dentro de TTDIoT, esto puede deberse a que dentro del conjunto de datos del que se aprendió esta estructura hay combinaciones que no se dieron. De los 60 valores dentro del conjunto de datos de prueba solo 14 combinaciones pueden encontrarse en la clasificación de [8]. De estos 14, solo 5 inferencias coincidieron con el enfoque algorítmico seleccionado.

```

"Multicriteria" => 35
"Heuristic" => 2
"Stochastic" => 9
"Supervised learning" => 14

```

Figure 4: Valores obtenidos con K2

4.2. Resultados con estructura obtenida con K2+Hill Climbing

4.2.1 DAG

El grafo obtenido de la ejecución del algoritmo K2+Hill Climbing (Fig. 5), presenta una relación de causalidad hacia el *Enfoque Algorítmico (Nivel II)*. A diferencia del grafo obtenido con K2, hay una relación causal de *Procesamiento* a *Objetivo*, la relación causal del *Objetivo* hacia la *Incertidumbre* se mantuvo. De acuerdo a esta estructura todas las variables tienen un efecto sobre la elección del *Enfoque Algorítmico (Nivel II)*. Esta estructura obtuvo como evaluación final $BD_{final} = -318.43$.

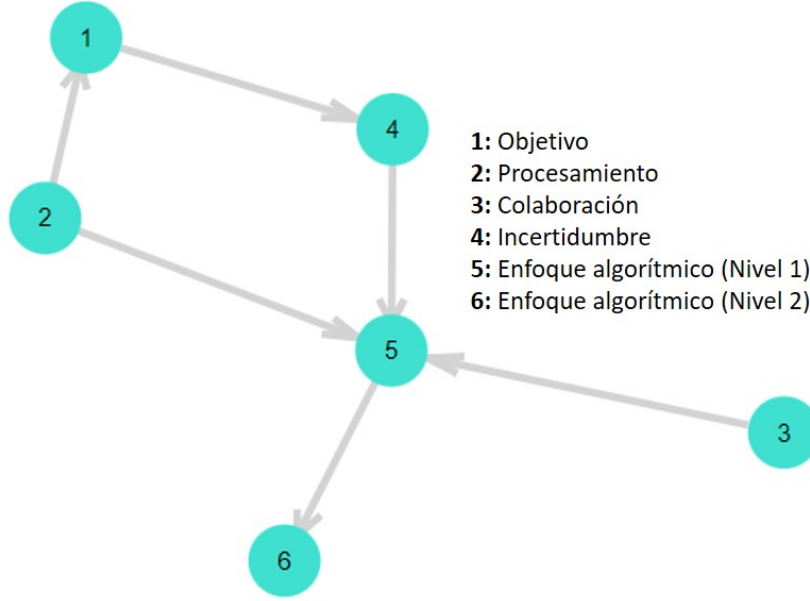


Figure 5: DAG obtenido con K2+Hil Climbing

4.2.2 Resultados inferencia

En la Fig. 6 se muestra la distribución de resultados inferidos. La estructura pudo inferir solo cuatro de las once técnicas dentro de TTDIoT, esto puede deberse a que dentro del conjunto de datos del que se aprendió esta estructura hay combinaciones que no se dieron. De los 60 valores dentro del conjunto de datos de prueba solo 14 combinaciones pueden encontrarse en la clasificación de [8]. De estos 14, solo 5 inferencias coincidieron con el enfoque algorítmico clasificado en [8].

```

"Multicriteria" => 28
"Heuristic" => 4
"Stochastic" => 12
"Supervised learning" => 16

```

Figure 6: Valores obtenidos con K2+Hil Climbing

4.3. Resultados con estructura obtenida con MWST+GPT Recovery+Hill Climbing

4.3.1 DAG

El grafo obtenido de la ejecución del algoritmo MWST+GPT Recovery+Hill Climbing (Fig. 7), presenta relaciones de causalidad diferentes a las obtenidos por los otros dos algoritmos. En esta estructura *Colaboración* tiene un efecto directo sobre *Enfoque Algorítmico (Nivel II)*, mientras que *Procesamiento* es afectado por *Colaboración*, *Enfoque Algorítmico (Nivel I)* e *Incertidumbre*. Es importante mencionar que MWST y GPT son método estadísticos, por lo que este grafo es más sesgado por las observaciones en el conjunto de datos de aprendizaje. Esta estructura obtuvo como evaluación final $BD_{final} = -398.88$.

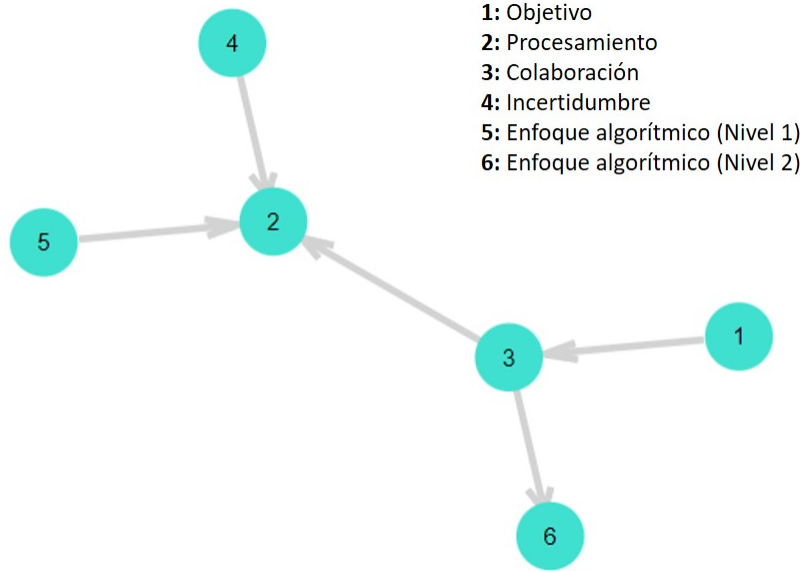


Figure 7: DAG obtenido con MWST+GPT Recovery+Hill Climbing

4.3.2 Resultados inferencia

En la Fig. 8 se muestra la distribución de resultados inferidos. La estructura pudo inferir solo dos de las once técnicas dentro de TTDIoT, esto se debe a que MWST y GPT son métodos que dan peso a los valores estadísticos. De los 60 valores dentro del conjunto de datos de prueba solo 14 combinaciones pueden encontrarse en la clasificación de [8]. De estos 14, solo 4 inferencias coincidieron con el enfoque algorítmico clasificado en [8].

```

"Multicriteria" => 23
"Supervised learning" => 37

```

Figure 8: Valores obtenidos con MWST+GPT Recovery+Hill Climbing

5. Conclusiones

En este trabajo se implementaron 3 algoritmos para el aprendizaje de una estructura Bayesiana a partir de un conjunto de datos clasificado con TTDIoT, posteriormente, las estructuras fueron utilizadas para inferir sobre un conjunto de datos de prueba.

De los tres algoritmos, K2+Hill Climbing obtuvo el mejor BD , sin embargo, esta mejora con respecto a la estructura obtenida con K2 fue muy pequeña.

Por su parte, la estructura obtenida con K2 se asemeja al mapeo de TTDIoT con los pasos de la resolución sistemática de problemas [8].

El algoritmo MWST+GPT recovery+Hill Climbing, pese a ser el más complejo, obtuvo el BD más bajo y la estructura menos esperada, esto se debe a que esta basado en estadísticos. En esta estructura, *Procesamiento*, que solo puede tener dos valores, resultó ser una variable sobre la que tienen efecto directo otras tres variables (*Colaboración*, *Incertidumbre* y *Enfoque Algorítmico (Nivel I)*). Algo que se exploró durante la implementación del algoritmo MWST+GPT recovery+Hill Climbing fue la prueba de independencia estadística χ^2 que mostró independencia entre las categorías *Incertidumbre* y *Colaboración*, este patrón de independencia se repite en las tres estructuras obtenidas.

En cuanto al desempeño en la inferencia, las tres estructuras presentaron un problema de sesgo causado por la falta de datos para determinar una distribución adecuada para cada posible de las variables aleatorias.

En conclusión, una estructura Bayesiana es un buen modelo para inferir una solución dado un vector de especificaciones pero encontrar un buen modelo requiere un conjunto de datos mayor.

References

- [1] Causal reasoning.
- [2] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, May 1968.
- [3] Ishita Dasgupta, Jane Wang, Silvia Chiappa, Jovana Mitrovic, Pedro Ortega, David Raposo, Edward Hughes, Peter Battaglia, Matthew Botvinick, and Zeb Kurth-Nelson. Causal Reasoning from Meta-reinforcement Learning. *arXiv:1901.08162 [cs, stat]*, January 2019. arXiv: 1901.08162.
- [4] Eugene Fink. Evaluation of representations in AI problem solving. In *2007 IEEE International Conference on Systems, Man and Cybernetics*, pages 349–353, Montreal, QC, Canada, October 2007. IEEE.
- [5] Jill H Larkin and Herberta Simon. Why a Diagram is (Sometimes)Worth Ten ThousandWords. page 36, 1987.
- [6] Tim A. Wheeler Mykel J. Kochenderfer and Kyle H. Wray. *Algorithmis for decision making*. MIT press, 2022.
- [7] Allen Newell and Herbert Alexander Simon. *Human problem solving*. Echo Point Books & Media, Brattleboro, Vermont, 2019.
- [8] Liliana Durán Polanco. Taxonomia para la toma de decisiones bajo incertidumbre en sistemas IoT. Master’s thesis, 2021.
- [9] George Rebane and Judea Pearl. The recovery of causal poly-trees from statistical data. In *Int. J. Approx. Reason.*, 1987.
- [10] Luis Enrique Sucar. *Probabilistic Graphical Models*. Advances in Computer Vision and Pattern Recognition. Springer London, London, 2015.

A. Apéndice A

El código se encuentra disponible en Jupyter Notebooks en Git Hub: https://github.com/LiliDupDev/Julia_DecisionMaking/tree/master/Bayesian%20structure%20learning.

- La libreta *Razonamiento causal para selección de algoritmo* contiene las implementaciones de los algoritmos de aprendizaje de estructura.
- La libreta *Inferencia* contiene las pruebas de inferencia.