



Ingeniería en Sistemas Computacionales

Minería de Grafos

Mapa mental de los algoritmos de caminos y búsqueda

Marco Ricardo Cordero Hernández

Tlaquepaque, Jal., 31 de noviembre de 2023

Algoritmos de caminos y búsqueda

Algoritmos de todos los caminos

Single Source Shortest Path

Funcionamiento
Desde un nodo inicial determinado, se selecciona la relación con el menor peso hacia otro nodo; después, se selecciona la siguiente relación con el menor peso *sumando el peso anterior* y se agrega a un árbol de decisiones. Al final, los caminos de menor costo dentro del árbol se seleccionan para obtener las rutas más cortas

Propósito
Calcular el camino más corto entre un único nodo raíz y *todos* los demás nodos

Cuándo usarlo
Cuando se desee evaluar la ruta más corta hacia múltiples destinos desde un solo origen, lo cual sería útil para servicios de emergencia, detectar cambios en una topología de red o dar sustento sistemas como LAN

Propósito
Encontrar el camino más corto entre todos los pares de nodos en un grafo

Funcionamiento
Calcula el camino más corto (con peso) entre todos los pares de nodos del grafo al "memorizar" las distancias calculadas hasta una iteración partiendo desde un nodo inicial. Estas distancias pueden ser usadas posteriormente para calcular el camino más corto hacia un nodo previamente no visitado.
Nota: los nodos no alcanzables no retornarán ningún valor

All Pairs Shortest Path

Propósito general
Encontrar los caminos más cortos entre múltiples nodos

Cuándo usarlo
Para optimizar la ubicación de servicios dentro de alguna ciudad, predecir la carga vehicular en ciertos tramos de un sistema de transporte, y encontrar una red con máximo ancho de banda y latencia mínima dentro de un centro de datos

Caso de uso
Evaluar rutas alternas para evitar un embotellamiento en el tráfico

Algoritmos de caminos más cortos

Propósito general
Encontrar el camino más corto entre un par de nodos

Funcionamiento
Usualmente, con la presencia de relaciones con peso es posible implementar el algoritmo de Dijkstra, en donde cada peso se va evaluando con cada nuevo nodo visitado, de forma que el algoritmo se considera acumulativo (más memoria para su ejecución)

Dijkstra

Caso de uso
Proporcionar direcciones de manejo desde un origen hacia un destino

Variación A*

Funcionamiento
Esta variación mejora el tiempo de ejecución de Dijkstra al preservar información adicional de los caminos parciales descubiertos al explorar los nodos y sus vecinos

Variación Caminos de Yen

Cuándo usarlo
Cuando se requiera encontrar direcciones entre dos ubicaciones o cuando se desee encontrar el grado de separación entre dos personas en una sociedad

Funcionamiento
Esta variación no solo calcula el primer camino más corto entre el origen y destino establecidos, sino que también calcula el segundo, tercero y hasta $k-1$ caminos más cortos. La utilidad de esto radica en la posibilidad de planes auxiliares cuando el camino inicial no resulta posible

Minimum Spanning Tree

Funcionamiento
Usando el algoritmo de Prim, el cual soporta pesos negativos, se toma un nodo inicial y se obtienen todos los nodos alcanzables desde el mismo, así como el conjunto de relaciones que los conecta con el menor peso posible.
A diferencia de Dijkstra, este algoritmo trata de minimizar la longitud de cada relación individual, en vez de la longitud total del camino entero.
El algoritmo es similar a *Single Source Shortest Path*, la diferencia radica en que la suma total de los pesos se ignora y solo se verifica el peso de las relaciones siguientes

Cuándo usarlo
Generalmente usado para encontrar la mejor ruta para visitar todos los nodos pero todos estos deben ser visitados sin omitir alguno, lo cual puede emplearse en sistemas de tuberías para distribución de agua o gas y posicionamiento de componentes dentro de circuitos electrónicos. La solución del famoso problema del viajante también puede aproximarse con este algoritmo, así como establecer un plan de reducción de costos para visitar un país o realizar un seguimiento histórico de la transmisión de una infección

Caso de uso
Optimizar ruteo en redes tanto en el aspecto físico para el posicionamiento de cables, como en el aspecto virtual para recolectores de basura (memoria)

Propósito
Calcular el camino en una estructura de árbol conexas con el menor costo de visita de todos los nodos

Random Walk

Propósito
Obtiene la lista de nodos existentes en un camino de tamaño n pre-determinado al escoger relaciones aleatorias para recorrer

Funcionamiento
También conocido como la caminata del borracho, este algoritmo conoce su origen y destino, pero la ruta que usa para trasladarse entre estos puede ser muy extensa. El algoritmo inicia en un nodo definido e intenta avanzar hacia otro nodo definido como destino, recorriendo un camino incierto a través de sus relaciones, pudiendo avanzar o retroceder con cada paso individual.
Nota: la aleatoriedad del algoritmo puede aumentar o disminuir de acuerdo al número de relaciones que un nodo y sus vecinos tiene

Cuándo usarlo
Como algoritmo complementario para flujos de datos cuando es necesario generar conjuntos aleatorios de nodos conectados. También puede usarse como parte del proceso de aprendizaje de modelos de machine learning e incluso para detección de comunidades.

Casos de uso
Aumentar el entrenamiento de un algoritmo de machine learning o generar datos para algoritmos de grafos

Conceptos clave

La base de la mayoría de los algoritmos de caminos se encuentra en la búsqueda de rutas entre un nodo de inicio y múltiples nodos intermedios hasta encontrar el destino deseado

Usados generalmente para logística, ruteo IP, reducción de costos, y más recientemente, simulación en videojuegos.

Caso de uso
Localizar lugares de interés cercanos en un sistema GPS

Depth First Search

Caso de uso
Encontrar el camino más corto en un árbol de decisiones en donde cada nivel afecta al subsecuente

Técnica de búsqueda que busca en *profundidad* a través de las estructuras, avanzando más y más a través de las relaciones de los nodos antes de pasar al vecino inmediato de un origen

Breadth First Search

Técnicas que dan sustento a muchos de los algoritmos de camino y búsqueda

Técnica de búsqueda que busca en *amplitud* a través de las estructuras, de forma que se busca en los vecinos inmediatos de los nodos primero