GRAFO DE K VECINOS MÁS CERCANOS

El artículo explora la construcción del grafo del k-vecino más cercano (KNNG) en datos de alta dimensión, destacando su uso en búsqueda de similitud, reducción de dimensiones y agrupamiento. Se centra en dos etapas principales del proceso: la construcción inicial del KNNG denominado INIT y su refinamiento mediante la propagación de vecindad denominado NBPG. El estudio presenta una comparación experimental de métodos existentes y propone alternativas para la inicialización del KNNG utilizando estructuras de indexación recientes como SW y HNSW. Además, se investiga la efectividad de la propagación de vecindad y su relación con las propiedades de los datos, concluyendo que ciertas características de los datos afectan significativamente el rendimiento de los métodos de propagación de vecindad. Los experimentos muestran que la elección del método inicial de KNNG y la estrategia de propagación de vecindad pueden tener un impacto considerable en la precisión y eficiencia del KNNG resultante. El estudio sugiere que la combinación de técnicas puede ofrecer mejoras significativas, especialmente en contextos donde la precisión y la eficiencia son críticas.

PALABRAS CLAVE: grafo de k vecinos más cercanos (KNNG), propagación de vecindad, datos de alta dimensión, concentración y análisis experimental

Metodología: El estudio se centra en dos etapas principales del proceso de construcción del KNNG: la inicialización (INIT) y la propagación del vecindario (NBPG). Se analiza la literatura existente, destacando la falta de comparaciones experimentales exhaustivas entre las soluciones propuestas y la ausencia de pruebas con estructuras de indexación modernas como SW y HNSW en la fase INIT.

Experimentación y Análisis:

- **Inicialización (INIT):** Se experimenta con métodos existentes y nuevas alternativas (SW y HNSW) para la creación inicial del KNNG, evaluando su impacto en la precisión del KNNG final.
- **Propagación de vecindad (NBPG):** Se investiga cómo la propagación de vecindad mejora la precisión del KNNG inicial, analizando la correlación entre las propiedades de los datos y la efectividad del NBPG.

Hallazgos Clave:

1. **Efectividad de Métodos INIT:** Los métodos modernos como HNSW muestran un rendimiento superior en la creación inicial del KNNG en comparación con métodos más tradicionales, ofreciendo un balance entre eficiencia y precisión.
2. **Impacto de la Propagación de vecindad:** La propagación de vecindad es crucial para refinar el KNNG, con una fuerte dependencia del tipo de datos. Los datos con alta "centralidad" o "hubness" tienden a beneficiarse menos de la propagación de vecindad
3. **Comparaciones Experimentales:** Se proporciona una evaluación comparativa integral de los métodos, destacando que no existe una solución única que domine en todos los escenarios. La elección del método adecuado depende específicamente de las características del conjunto de datos y los requisitos de la aplicación.

Recomendaciones:

- Implementar HNSW para INIT cuando se busca un balance óptimo entre eficiencia y precisión.
- Considerar la característica de "hubness" de los datos antes de elegir la estrategia de NBPG.
- Realizar evaluaciones comparativas en contextos específicos de aplicación para determinar la mejor combinación de métodos INIT y NBPG.

Conclusión: El estudio ofrece insights prácticos y metodológicos para mejorar la construcción de KNNG en datos de alta dimensión, con recomendaciones específicas para maximizar la eficacia de las técnicas utilizadas. La adaptabilidad de las estrategias en función de las propiedades de los datos es crucial para optimizar el rendimiento y la aplicabilidad del KNNG en diversas aplicaciones.

El grafo de (KNNG) se utiliza en → visión artificial, minería de datos, Aprendizaje automático

KNNG → Dado un conjunto de datos de dimensión d: $D \subseteq \mathbb{R}^d$ y un entero positivo K trata cada punto de $U \in D$ como un nodo del grafo y crea aristas desde U hasta sus K vecinos más cercanos (KNN) en $D \setminus \{u\}$

En este artículo utilizaron punto, vector y nodo de manera intercambiable.

(KNNG) se utiliza para varias operaciones:

* Búsqueda aproximada de k vecinos más cercanos (ANN)
* Los métodos de búsqueda de (ANN) de última generación como → DPG, NSG

Para ello primero construyen (KNNG) preciso en D y luego ajustan el vecindario para cada nodo en la parte superior de (KNNG)

MÉTODO

Construcción Rápida (luego se refina)
de un KNNG inicial con precisión baja o moderada
Se emplean ESTRUCTURAS de INDICE
- Árbol de partición binaria
- Árbol de proyección aleatorio
- hash sensible a la localidad

Se denota como INIT
adoptaron 2 técnicas relacionadas para la búsqueda de (ANN)
SW KNNG, HNSW KNNG

Propagación de vecindad
Los métodos existentes siguen el mismo principio "un vecino de mi vecino también es probable que sea mi vecino."

Se denota como NBPG
Clasificaron los métodos en 3 categorías
Uniprop
solo considera los vecinos más cercanos en NBPG
Según los autores no logra KNNG de alta precisión y es sensible a K
Biprop
adoptan el enfoque iterativo para refinar el KNNG (inicial) paso a paso
considera tanto a los vecinos más cercanos como a los vecinos más cercanos inversos
tiene mejor rendimiento para memoria
Deep Search
solo se ejecuta para realizar búsquedas de (ANN) en un grafo de proximidad en línea construido

Eficacia
depende de la propiedad de los datos
Concentración de nodos de un nodo → Se define → Nº de sus KNN inversas exactas
concentración de datos de un conjunto de datos → Se define → suma normalizada de la concentración de nodos más grande de sus nodos superiores hasta un porcentaje especificado, con el fin de caracterizar un conjunto de datos

HALLAZGOS
- Datos con concentración alta suelen tener un peor rendimiento de NBPG que unos datos de concentración baja
- Un nodo con concentración alta sólo tiene una mayor precisión de sus KNN que un nodo con concentración baja

Deep HNSW
es una nueva combinación que presenta el mejor equilibrio entre EFICIENCIA, PRECISIÓN y MEMORIA en la mayoría de los casos

PROBLEMA DE INVESTIGACION

Sean $D \subseteq \mathbb{R}^d$ un conjunto de datos que consta de n d-dimensionales vectores reales. El KNNG en D se define como

Def1. KNNG Sea $D \subseteq \mathbb{R}^d$ y un entero positivo K, Sea $G = (V, E)$ el KNNG de D donde V es el conjunto de nodos y E es el conjunto de aristas. Cada nodo $v \in V$ representa de manera única un vector en D. Existe una arista dirigida $(u, v) \in E$ si y solo si V es uno de los KNN de U en $D \setminus \{u\}$

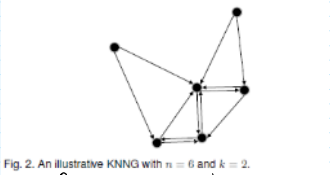

Fig. 2. An illustrative KNNG with $n = 6$ and $k = 2$.

En la figura se muestra el KNNG de un conjunto de

Fig. 2. An illustrative KNNG with $n = 6$ and $k = 2$.

En la figura se muestra el KNNG de un conjunto de datos con 6 puntos de datos. Para cada nodo, tiene 2 aristas salientes que apuntan a sus 2NN

El artículo se centra en: soluciones en memoria sobre vectores densos de alta dimensión con la distancia euclidiana como medida.

→ Marco de Construcción

Algoritmo 1: Marco de construcción de KNNG

Entrada: Conjunto de datos D y un entero positivo k
Salida: KNNG G
1 INIT: generar un gráfico inicial G rápidamente;
2 NBPG: refinar G por propagación de vecindad; 3 devolver G;

También analizamos sus requerimientos de memoria y complejidad temporal. Para los requerimientos de memoria, solo contamos las estructuras de datos auxiliares de cada método, ignorando las estructuras comunes, es decir, los datos D y el KNNG G.

→ KNNG Vs Grafo de Proximidad

3

Algoritmo 2: Búsqueda en el gráfico (H, q, k, efSearch, ep)

Entrada: H, q, k, efSearch y ep
Salida: KNN de q 1, deje
que el grupo sea el conjunto candidato e inserte ep en el grupo;
2 L = max(k, efSearch) e i = 0; 3 mientras i <
efSearch haga 4 u = pool[i] y
marque u como expandido;
5 /* Procedimiento: expand(q, u, H)*/ para
6 cada vecino v de u en H hacer
7 /* Procedimiento: update(pool, v)*/
8 pool.add(v, dist(q, v)); ordenar
9 pool en orden ascendente de dist(q, ·); si pool.size()
10 > L, entonces pool.resize(L);
11 i = índice del primer nodo no expandido u en el grupo;
12 devuelve los primeros k nodos del grupo;

→ Búsqueda de (ANN)

- Sea H un grafo de proximidad y q una consulta. La búsqueda comienza en uno o más nodos ep específicos o seleccionados aleatoriamente, que primero se introduce en un conjunto de candidatos (una lista de nodos ordenada)
- Sea L = max(K, efSearch) el tamaño de pool. En cada iteración de la línea 4-11, encontramos el primer nodo no expandido u en pool[L] y luego expandimos u en las líneas 6-10 lo que se denota como expand(q, u, H). Esta expansión trata a cada vecino v de U en H como un candidato para refinar el pool en la línea 8-10, lo que se denota como update(pool, v). Una vez que se han expandido los primeros nodos efSearch en pool, el proceso de búsqueda finaliza y devuelve los primeros K nodos en el pool. Obviamente efSearch es clave para el rendimiento de la búsqueda. Un efSearch grande aumenta tanto el costo como la precisión

(3) → Construcción de Grafos Iniciales

Métodos para INIT

Enfoque basado en Particiones
↓
particionar los puntos de datos en grupos suficientemente pequeños pero no similares
↓
Luego cada punto encuentra sus KNN en el grupo o grupos más prometedores y se genera un KNNG inicial
↓
Incluye → La división múltiple LSH KNNG
→ Faiss Vis
Métodos q' adoptan diferentes técnicas de partición

Enfoque basado en mundos pequeños
↓
Los puntos de datos se dividen en grupos lo suficientemente pequeños pero similares entre sí
↓
Consiste en crear un gráfico de mundo pequeño navegable. enhe los puntos de datos, cuya estructura es útil para identificar los KNN de cualquier punto de datos
↓
técnicas
SW KNNG — HNSW KNNG

→ División Aleatoria múltiple → wang et al → propusieron este enfoque. para construir grafos de vecindad aproximados de base y luego unirlos para obtener un KNNG inicial.

- Particiona los puntos de datos en D en 2 grupos disyuntos.
- Para asignar puntos cercanos al mismo grupo, la partición se realiza a lo largo de la dirección principal.
- El conjunto de datos D se particiona en 2 grupos disyuntos D1 y D2 que se dividen independientemente de manera recursiva hasta que el grupo a dividir contiene como máximo T_{div} puntos. Luego se construye un subgrafo de vecindad para cada grupo $O(d * T_{div}^2)$

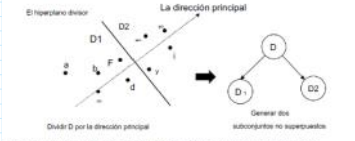

Fig. 3. Ilustre la división de D por la dirección principal (representada por la flecha continua). Como se muestra con la línea discontinua, el hiperplano de división que es perpendicular a la dirección principal divide a D en dos grupos disjuntos, D1 y D2, que se dividen de forma recursiva. La figura debe recrearse.

Una única división produce un grafo de vecindad aproximado base que contiene una serie de subgráficos aislados y no puede conectar un punto con sus vecinos más cercanos que se encuentran en subgrafos diferentes. Por lo tanto la división múltiple utiliza divisiones aleatorias L_{div} que permite calcular la dirección principal a partir del conjunto grupo Para cada punto $U \in D$ tiene un conjunto de KNN de cada división y por lo tanto los conjuntos L_{div} totales de KNN.

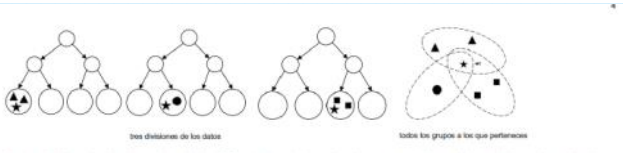

Fig. 4. Esta figura ilustra tres divisiones aleatorias de un conjunto de datos D. Para el punto de datos u denotado como una estrella, sus KNN se calculan en el grupo de cada uno. división y luego se combinaron para devolver los mejores KNN.

Cost Analysis

Dado que esas divisiones I_{div} se pueden generar de forma independiente, el requisito de memoria es $O(n)$ donde se almacena la división actual. Para cada división, la función recursiva partition cost $O(n*d*\log n)$ y el procedimiento de brute-force en todos los grupos finalmente cons ya dos, se requiere $O(n*d*T_{div})$. (log I_{div} divisiones +the cost of multiple división is $O(I_{div}*n*d*(\log n + T_{div}))$ Podemos ver que I_{div} y T_{div} son clave. para cost of multiple division.

3.2 LSH KING → Zhang et al propusieron el método

Utiliza la técnica de hash para dividir D en grupos de igual tamaño y luego construye un grafo de vecindad en cada grupo usando el método brute-force. Se crean divisiones I_{hash} para construir la base I_{hash} de gráficas de vecindad aproximadas que luego se combinen para generar un KNNG de mayor precisión.

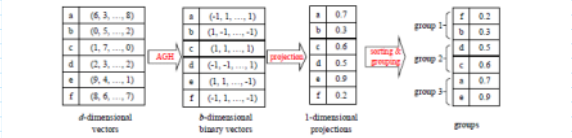

Fig. 5. The process of creating a division by LSH in LSH KNNG.