VŠB TECHNICKÁ UNIVERZITA OSTRAVA
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Vyhledávání K nejbližších sousedů na základě filtru

Search K nearest neighbors based on a filter

Bc. Jan Jedlička

Vedoucí práce: Doc. Ing. Radim Bača, Ph.D.

Ostrava, 2022

Abstrakt

Techniky pro efektivní vyhledání K nejbližších sousedů (tzv. KNN problém) jsou základem pro mnoho dnešních aplikací. Velmi často se využívají i techniky pro přibližné KNN vyhledávání. Tyto techniky jsou založeny na grafech. Předmětem této práce rozšíření existující implementace pro přibližné KNN vyhledávání o možnost specifikovat filtr. Filtr bude podmínka, která stanoví, které vektory se při prohledávání vynechají.

Klíčová slova

KNN;HNSW;Filter

Abstract

Techniques for effective nearest neighbor search (so-called KNN problem) are the basis for many of today's applications. Techniques for approximate KNN searches are also very often used. These techniques are based on graphs. The subject of this work is to extend the existing implementation for approximate KNN searches with the ability to specify a filter. The filter will be a condition that determines which vectors are omitted when searching.

Keywords

KNN;HNSW;Filter

Poděkování

Rád bych na tomto místě poděkoval vedoucímu semestrálního projektu, kterým byl pan Doc. Ing. Radim Bača, Ph.D., za pravidelné konzultace a poskytnutí mnoha užitečných rad a nápadů pro řešení samotné práce.

Obsah

Seznam použitých zkratek a symbolů

KNN	–	K-nearest neighbors
HNSW	–	Hierarchical Navigable Small Worlds

Kapitola 1

Úvod

Cílem této semestrální práce bylo pochopit a následně naimplementovat HNSW algoritmus vyhledávající K nejbližších sousedů v prostoru n-diemnzionálních vektorů. Následně bylo zapotřebí tuto mou implementaci rozšířit o vyhledávání prvků na základě zadaných filtrů. Filtry jsou booleovsé podmínky omezující hodnoty jednotlivých atributů u prvků pro K vyhledaných sousedů. Veškerá práce byla napsána v jazyku C++. Tento jazyk je volen vzhledem k jeho nízké úrovni, což umožňuje maximální rychlosti, přímou práci s pamětí, a tato vlastnost je zvlášť u databázových operací a systémů podstatná.

Kapitola 2

KNN

KNN algoritmy slouží pro získání K nejbližších prvků od zvoleného prvku (QueryNode/TargetNode) v n-dimenzionální prostoru. Tyto techniky jsou základem pro mnoho dnešních aplikací a často se i využívají metody pro přibližné KNN vyhledávání. Tato hodnota K bývá relativně malá, nejčastěji získáváme 10 nejbližších prvků, případně hodnoty pod 100, s tím že K může nabývat i mnohonásobně vyšších hodnot což ale není příliš časté.

Pro určení vzdálenosti mezi jednotlivými prvky v n-dimenzionálním prostoru můžeme využít celou řadu metrik. Mezi ty nejznámější patří Euklidova, Hammingova, Minkovského případně Čebyševova. Ve vypracované implementaci se vždy pracuje s Euklidovou vzdáleností, nicméně by nebyl problém zaměnit definici metody pro určení vzdálenosti mezi prvky a tím pádem změnit použitou metriku.

Dimenze prostoru ve kterém vyhledáváme prvky nemusí být vůbec nízká. Dimenze prostoru se může pohybovat v rozmezí od jednotek (2,5,...) do stovek (sift vecdim = 128). A hodnoty jednotlivých atributů se mohou rovněž tak pohybovat v širokém spektru, s tím že v mé práci jsou všechny atributy datového typu float.

Jako datový soubor pro vytvoření bodů v prostoru a možnosti následného testování byl vybrán Sift1M. V souboru se nachází binární data obsahující 1 000 000 rozdílných bodů, v prostoru o dimenzi 128 a floatové hodnotě jendotlivých atributů v rozmezí od 0 do 217, přičemž v atributech jsou pouze celočíselné hodnoty.

2.1 HNSW

2.2 Implementace HNSW

Pro lepší pochopení HNSW se již naimplementovaný projekt využíval pouze jako reference pro porovnání posobnosti výsledků a srovnání časů vykonání operací. Implemence HNSW se tedy vytvořila od základu nová, což přineslo výhodu v maximálním porozumění vlastnímu kódu. Na druhou stranu

je tato implementace přibližně 2.5x pomalejší než referenční kód, výsledky ale vrací rovněž validní jako reference.

2.3 Filtr

Pod pojmem filtr si můžeme představit booleovskou podmínku omezující hodnoty jednotlivých atributů (dimenzí) prvků v n-dimenzionálním prostoru. Konkrétně tedy filtr říká jaké hodnotě se mají jednotlivé atributy rovnat, případně v jakém intervalu by se měly atributy pohybovat. Toto platí pouze pro atributy jež filtr omezuje, atributy které filtr nezmiňuje vůbec nekontrolujeme a akceptujeme je bez závislosti na jejich hodnotě.

U KNN slouží filtr pro omezení vyhledání výsledných K prvků. Při vyhledávání výsledku se prochází i prvky které filtru nevyhovují, ale do vráceného výsledku jsou vloženy pouze prvky jejichž atributy vyhovují omezení filtru.

Selektivita filtru je číslo v rozsahu od 0 do 1 a udává procentuální počet prvků, které filtr přijme. Čím více je filtr vybíravý tím bližší hodnota k 0 mu bude přiřazena a tím více je filtr selektivnější. Naopak filtry přijímající většinu prvků budou mít přiřazeny číslo blíže k 1 a jejich selektivita bude tedy klesat, s tím že filtry přijímající úplně všechny prvky budou mít hodnotu rovno 1.

2.3.1 Implementace filtru

2.3.2 Použití filtru v HNSW

2.3.3 Problém filtru s vysokou selektivitou u HNSW

Při použití implementace HNSW s využitím filtru se může stát že nám metoda KNNFilter nevrátí K hledaných prvků, ale výsledný počet hledaných prvků bude nižší. Pokud tak nastane tak nám ve výsledku chybí nízky počet prvků, nejčastěji 1 nebo 2 a pravděpodobnost že k tomuto jevu dojde je relativně nízká.

Tento jev nastává ve chvíli kdy hodnota K je stejně vysoká nebo jen o něco málo nižší než EF a zároveň je selektivita filtru vyšší, do 25%.

K tomuto problému nastává protože HNSW algoritmus pro procházení grafu a výběr nalezených prvků použitý v metodě SearchKNNFilter lze zakončit dvěma způsoby. První způsob jak zakončit algoritmus průchodu který nás aktuáně nezajímá je ten kdy vzdálenost nejbližšího prvku z C je vyšší než vzdálenost nejvzdálenějšího prvku z W a zároveň je počet prvků v F (původně W) roven zadanému K.

Druhý způsob jak ukončit průchod, ten který způsobí problém, je ten kdy je C prázdný. Stane se tedy že projdeme prvky z C a všechny jejich nenavštívené sousedy kteří jsou blíže než nejvzdálenější prvek z W. Během tohoto průchodu jednoduše nenarazíme na dostatečný počet prvků které by vyhovovaly filtru a F obsahuje tedy méně nejbližších prvků než je počet co hledáme.

Tento problém lze jednoduše vyřešit tím že zvýšíme hodnotu Ef aby byla více rozdílna a vyšší než hodnota K. S roustoucím Ef při zachování K a filtru poroste i čas vykonání operace, ale bude se zvyšovat pravděpodobnost že získáme přesně K prvků ve výsledku a poroste i přesnost operace.

Kapitola 3

Závěr

složitější debug pro velké grafy a random vkládání do layers, také když se 2 body mají stejnou vzdálenost tak může dojít k rozdílu

Literatura

1. *ann-benchmarks* [online]. 2022. [cit. 2022-03-06]. Dostupné z: `http://ann-benchmarks.com/index.html`.
2. *git-hnswlib: hnswlib* [online]. 2022. [cit. 2022-03-06]. Dostupné z: `https://github.com/nmslib/hnswlib`.
3. *Nearest neighbor search* [online]. 2022. [cit. 2022-03-06]. Dostupné z: `https://en.wikipedia.org/wiki/Nearest_neighbor_search`.
4. *Optimalizace v INFORMIXU* [online]. 2022. [cit. 2022-04-04]. Dostupné z: `http://www.ms.mff.cuni.cz/~jkoc5219/Optimalizace_v_INFORMIXU.html`.