

Vyhledávání K nejbližších sousedů na základě filtru

Search K nearest neighbors based on a filter

Bc. Jan Jedlička

Vedoucí: Doc. Ing. Radim Bača, Ph.D.

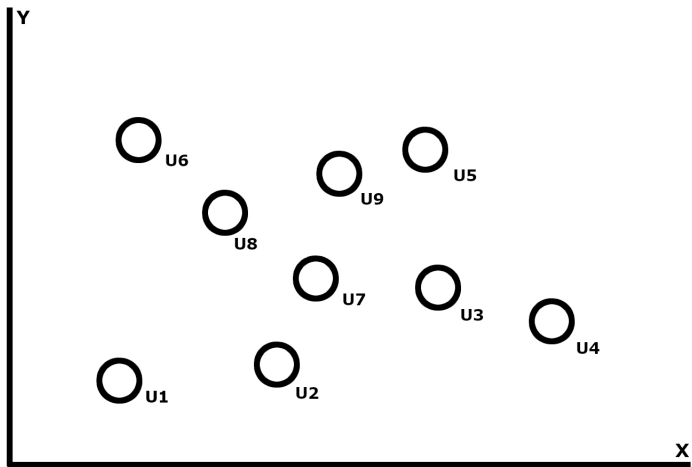
FEI, VŠB-TUO

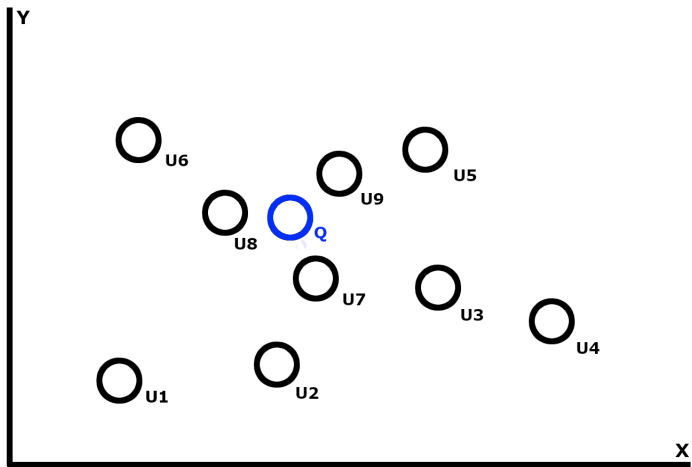
2022

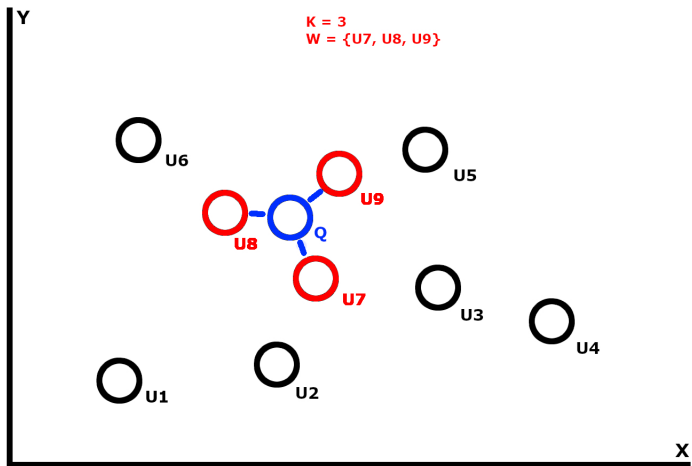


- Porozumění HNSW
- Vlastní HNSW implementace nebo zprovoznění jiné HNSW implementace
- Návrh a implementace rozšíření HNSW o filtr (podmínka, která stanoví, které vektory se při prohledávání vynechají)

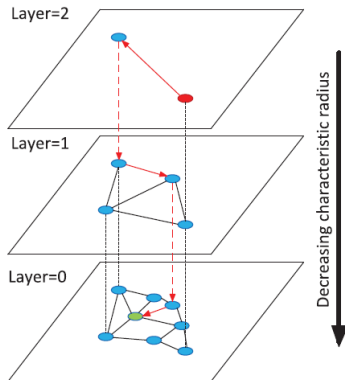
- Vyhledávání K nejbližších sousedů od dotazu Q v n dimenzionálním prostoru
- Vzdálenost mezi body v prostoru definována metrikou (Euklidova, Hammingova, Minkowského atd.)
- U velkých dimenzí je pro většinu technik rychlejší sekvenční průchod
- Přibližné vyhledávání (ANN)
- Porovnávání vektorizovaných dat, hledání shluků, podobných vlastností (například vyhledávání sémanticky podobných dokumentů)







- Hierarchical Navigable Small Worlds
- Řešení KNN problému, přibližné vyhledávání s využitím vícevrstevných grafů
- Výsledek poskytován s určitou přesností zvanou Recall
- Přesnost se dá zvýšit navýšením hodnoty parametru E_f , stejně tak poroste ale i čas vykonání dotazu



Obrázek: Vrstvy grafů v HNSW

Algorithm 2. SEARCH-LAYER(q, ep, ef, l_c)

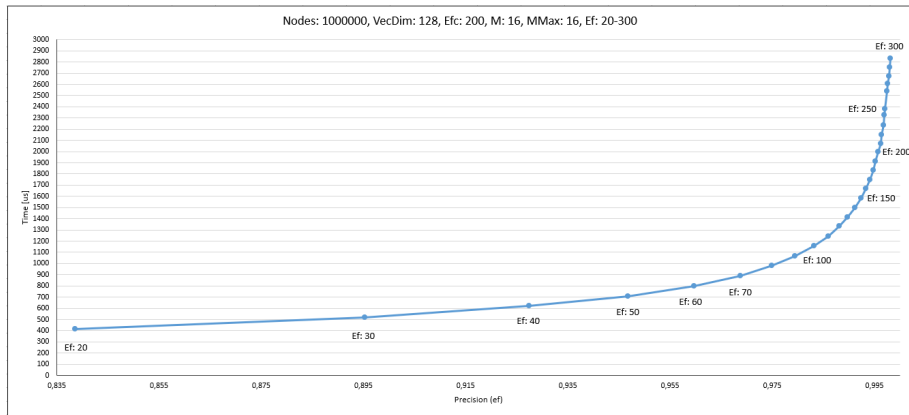
Input: query element q , enter-points ep , number of nearest to q elements to return ef , layer number l_c

Output: ef closest neighbors to q

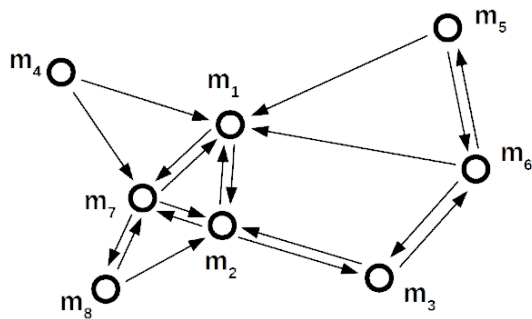
```

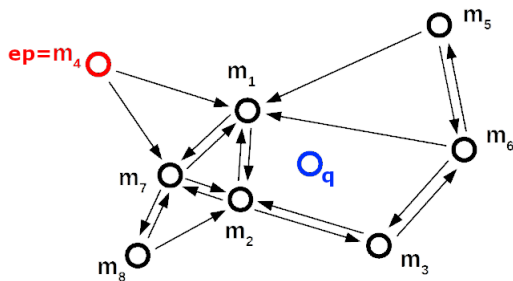
1  $v \leftarrow ep$  // set of visited elements
2  $C \leftarrow ep$  // set of candidates
3  $W \leftarrow ep$  // dynamic list of found nearest neighbors
4 while  $|C| > 0$ 
5    $c \leftarrow$  extract nearest element from  $C$  to  $q$ 
6    $f \leftarrow$  get furthest element from  $W$  to  $q$ 
7   if  $distance(c, q) > distance(f, q)$ 
8     break // all elements in  $W$  are evaluated
9   for each  $e \in neighbourhood(c)$  at layer  $l_c$  // update  $C$  and  $W$ 
10    if  $e \notin v$ 
11       $v \leftarrow v \cup e$ 
12       $f \leftarrow$  get furthest element from  $W$  to  $q$ 
13      if  $distance(e, q) < distance(f, q)$  or  $|W| > ef$ 
14         $C \leftarrow C \cup e$ 
15         $W \leftarrow W \cup e$ 
16      if  $|W| > ef$ 
17        remove furthest element from  $W$  to  $q$ 
18 return  $W$ 
```

Obrázek: Pseudokód HNSW Search algoritmu



Obrázek: Graf závislosti průměrného času jednoho dotazu na přesnosti



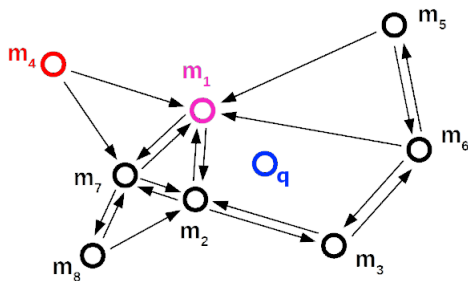


- $ep = \{m_4\}$

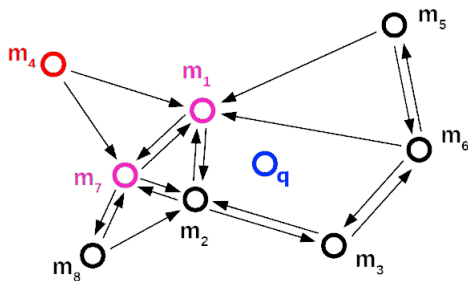
- $V = \{m_4\}$

- $W = \{m_4\}$

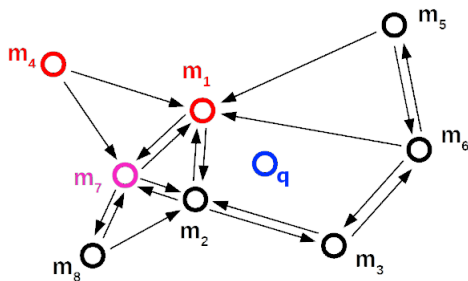
- $C = \{m_4\}$



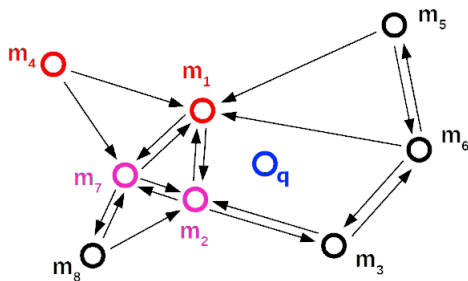
- $V = \{m_4, m_1\}$
- $W = \{m_1, m_4\}$
- $C = \{m_1\}$
- $f = m_4$
- $c = m_4$



- $V = \{m_4, m_1, m_7\}$
- $W = \{m_1, m_7, m_4\}$
- $C = \{m_1, m_7\}$
- $f = m_4$
- $c = m_4$



- $V = \{m_4, m_1, m_7\}$
- $W = \{m_1, m_7, m_4\}$
- $C = \{m_7\}$
- $f = m_4$
- $c = m_1$



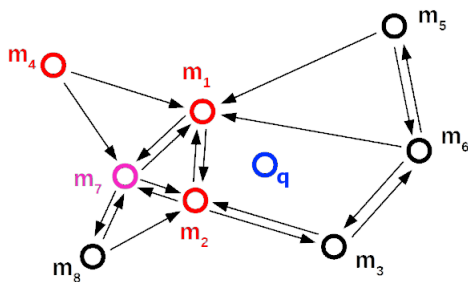
- $V = \{m_4, m_1, m_7, m_2\}$

- $W = \{m_1, m_2, m_7\}$

- $C = \{m_2, m_7\}$

- $f = m_7$

- $c = m_1$



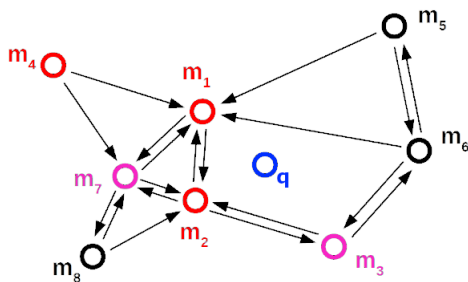
- $V = \{m_4, m_1, m_7, m_2\}$

- $W = \{m_1, m_2, m_7\}$

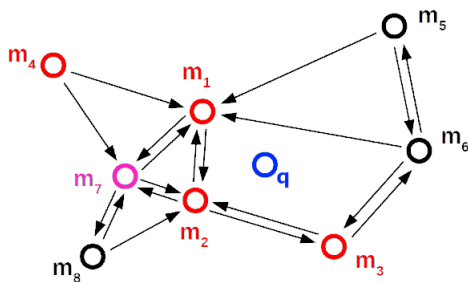
- $C = \{m_7\}$

- $f = m_7$

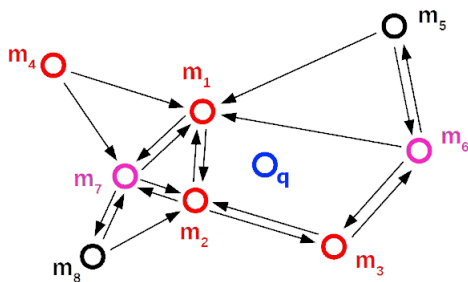
- $c = m_2$



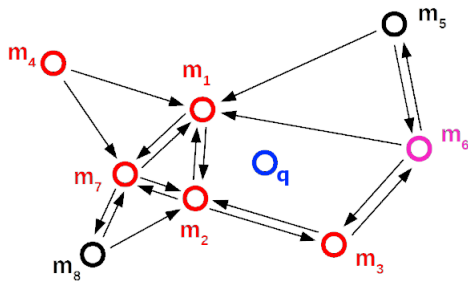
- $V = \{m_4, m_1, m_7, m_2, m_3\}$
- $W = \{m_1, m_2, m_3\}$
- $C = \{m_3, m_7\}$
- $f = m_3$
- $c = m_2$



- $V = \{m_4, m_1, m_7, m_2, m_3\}$
- $W = \{m_1, m_2, m_3\}$
- $C = \{m_7\}$
- $f = m_3$
- $c = m_3$



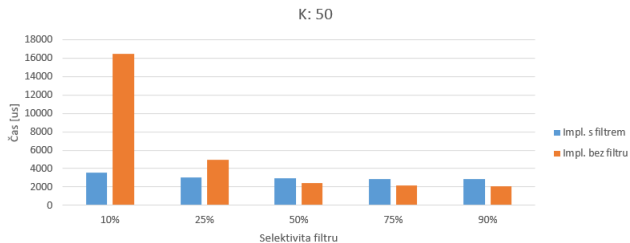
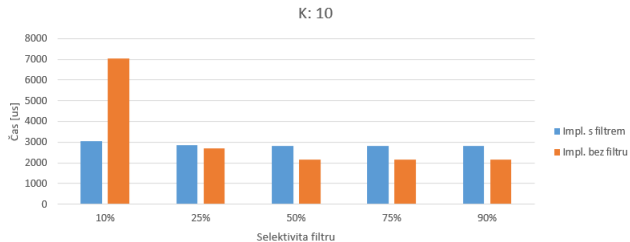
- $V = \{m_4, m_1, m_7, m_2, m_3, m_6\}$
- $W = \{m_1, m_2, m_3\}$
- $C = \{m_7, m_6\}$
- $f = m_3$
- $c = m_3$

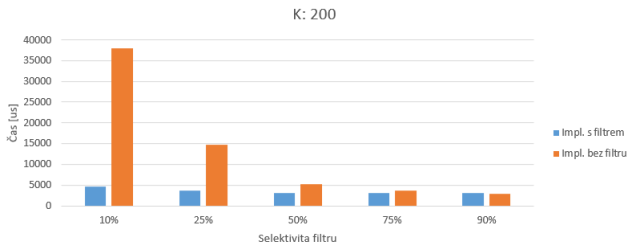
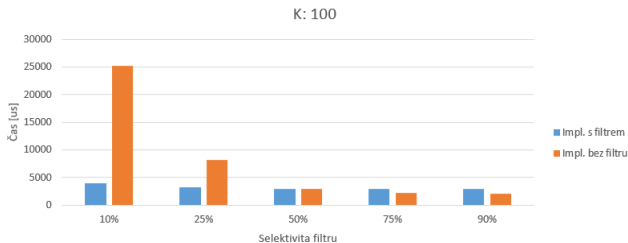


- $V = \{m_4, m_1, m_7, m_2, m_3, m_6\}$
- $W = \{m_1, m_2, m_3\}$
- $C = \{m_6\}$
- $f = m_3$
- $c = m_7$
- $dist(c, q) > dist(c, f)$

- Podmínka určující které vektory neprocházet a nevracet do výsledku
- Udává jakých hodnot mají nabývat jednotlivé atributy, nebo v jakých intervalech hodnot se mají nacházet
- Nemusíme omezovat všechny atributy
- Selektivita filtru $< 0, 1 >$ udává procentuální počet uzlů z celé množiny všech uzlů, které filtr přijme

- 72: $\langle 51.32, 143.87 \rangle$; 88: $\langle 3 \rangle$; 110: $\langle 72.40, 106.84 \rangle$;
- $\text{vec}[72] \in \langle 51.32, 143.87 \rangle$
- $\text{vec}[88] = 3$
- $\text{vec}[110] \in \langle 72.40, 106.84 \rangle$





- Splnění všech požadavků
- Funkční implementace původního HNSW
- HNSW implementace o polovinu pomalejší než reference
- Funkční implementace rozšířeného HNSW o filtry

Děkuji za pozornost



ann-benchmarks [online]. 2022. [cit. 2022-03-06]. Dostupné z: <http://ann-benchmarks.com/index.html>.



git-hnswlib: hnswlib [online]. 2022. [cit. 2022-03-06]. Dostupné z: <https://github.com/nmslib/hnswlib>.



git-hnsw: hnswlib [online]. 2022. [cit. 2022-04-10]. Dostupné z: <https://github.com/RadimBaca/HNSW>.



Nearest neighbor search [online]. 2022. [cit. 2022-03-06]. Dostupné z: https://en.wikipedia.org/wiki/Nearest_neighbor_search.



Optimalizace v INFORMIXU [online]. 2022. [cit. 2022-04-04]. Dostupné z: http://www.ms.mff.cuni.cz/~jkoc5219/Optimalizace_v_INFORMIXU.html.



MALKOV, Yu A; YASHUNIN, Dmitry A. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*. 2018, roč. 42, č. 4, s. 824–836.



Metrický prostor. 2022. Dostupné také z: https://cs.wikipedia.org/wiki/Metric%C3%BD_prostor#Definice.



K-Nearest Neighbors (KNN) algorithm. 2022. Dostupné také z: <https://towardsdatascience.com/k-nearest-neighbors-knn-algorithm-23832490e3f4>.