Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs

Yu. A. Malkov, D. A. Yashunin

IEEE Transactions on Pattern Analysis and Machine Intelligence 2018 (IEEE PAMI 2018)

= Approximate KNN (ANN)

Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs

= HNSW; Method

Yu. A. Malkov, D. A. Yashunin

IEEE Transactions on Pattern Analysis and Machine Intelligence 2018 (IEEE PAMI 2018)

Index

- Intro
- Background
- Method
- Experiments
- Conclusion

Intro

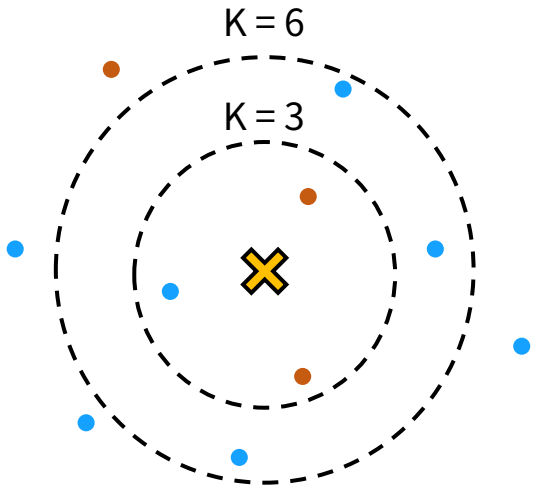
= Approximate KNN (ANN)

Efficient and robust approximate nearest neighbor search

using Hierarchical Navigable Small World graphs

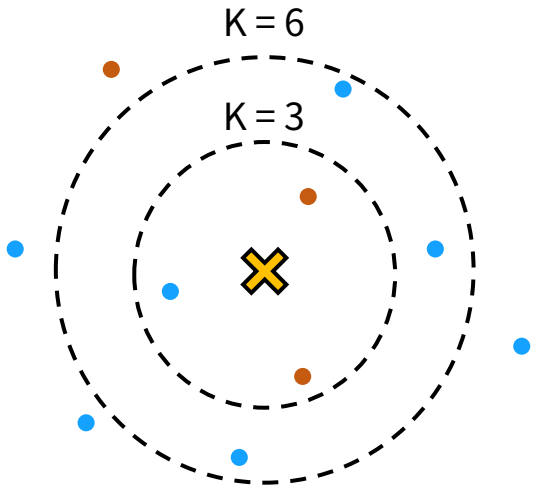
= HNSW; Method

Intro : Approximate Nearest Neighbor (ANN)



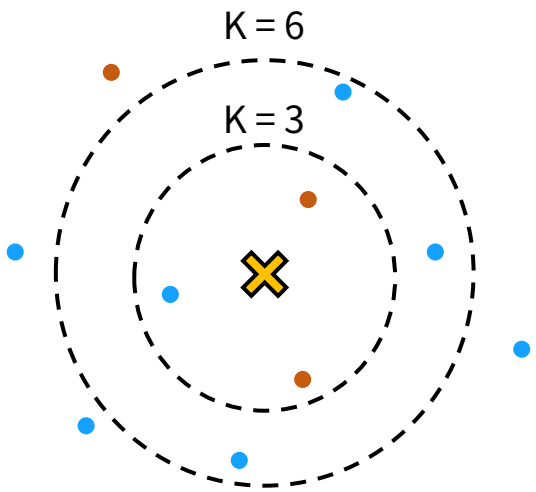
- K-Nearest Neighbor Search (KNN Search)
  - 쿼리 아이템과 가장 가까운(=유사한) K개의 이웃 찾기
  - 일반적으로 임베딩과 함께 사용

Intro : Approximate Nearest Neighbor (ANN)



- KNN의 사용 예시
  - 가장 가까운 아이템들의 라벨 참고 (Classification)
  - 가장 가까운 아이템들의 임베딩 가져오기 (RAG 등)

Intro : Approximate Nearest Neighbor (ANN)



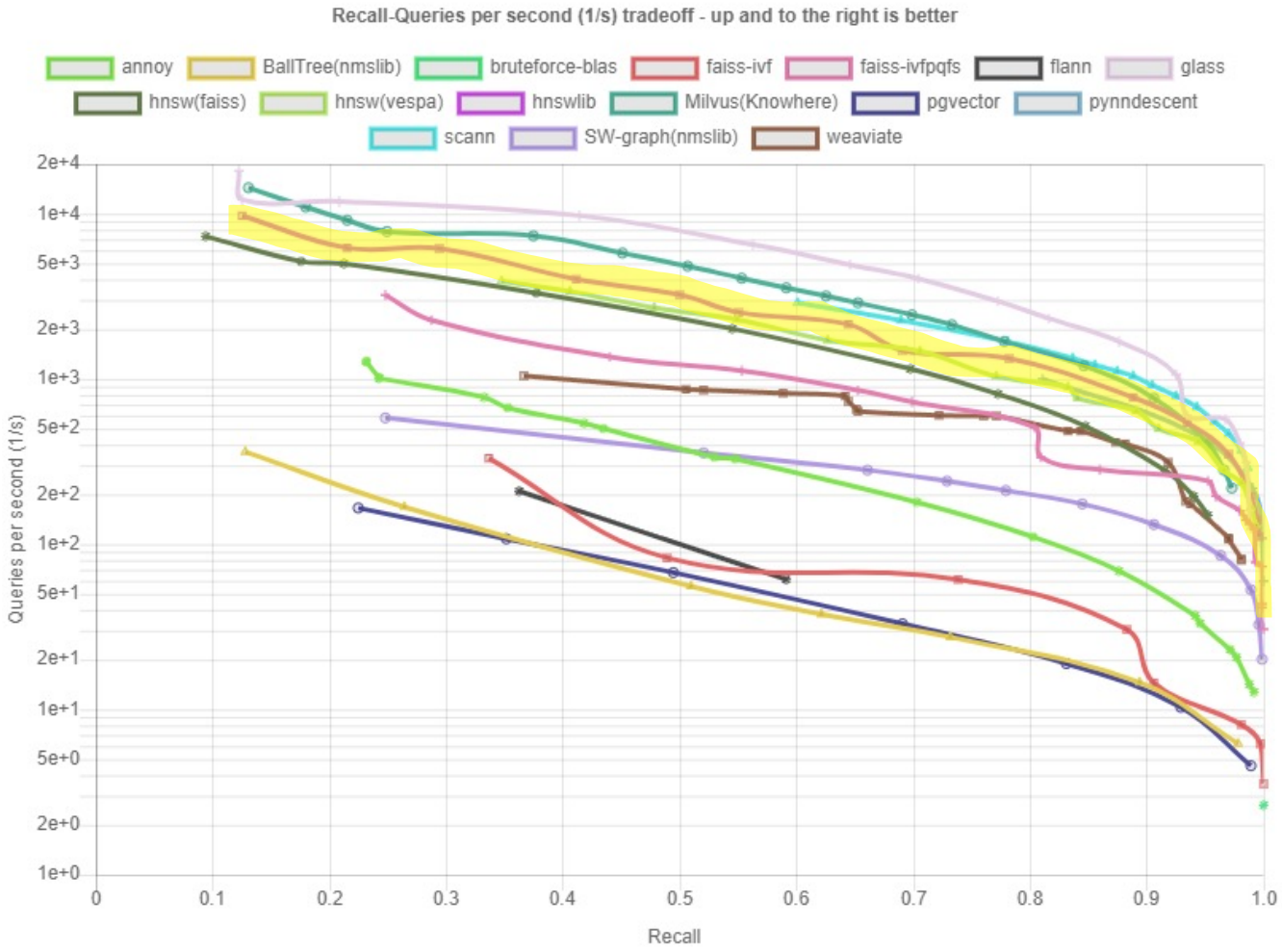
- 효과적인 KNN Search
  - Naïve한 접근 (Brute force):
    - 모든 아이템과의 거리 계산
    - O(N) -> 데이터가 커질수록 엄청나게 느려짐

→ 효율적인 방법들은 O(log N)에 검색!

→ Approximate한 접근을 통해 성능 향상

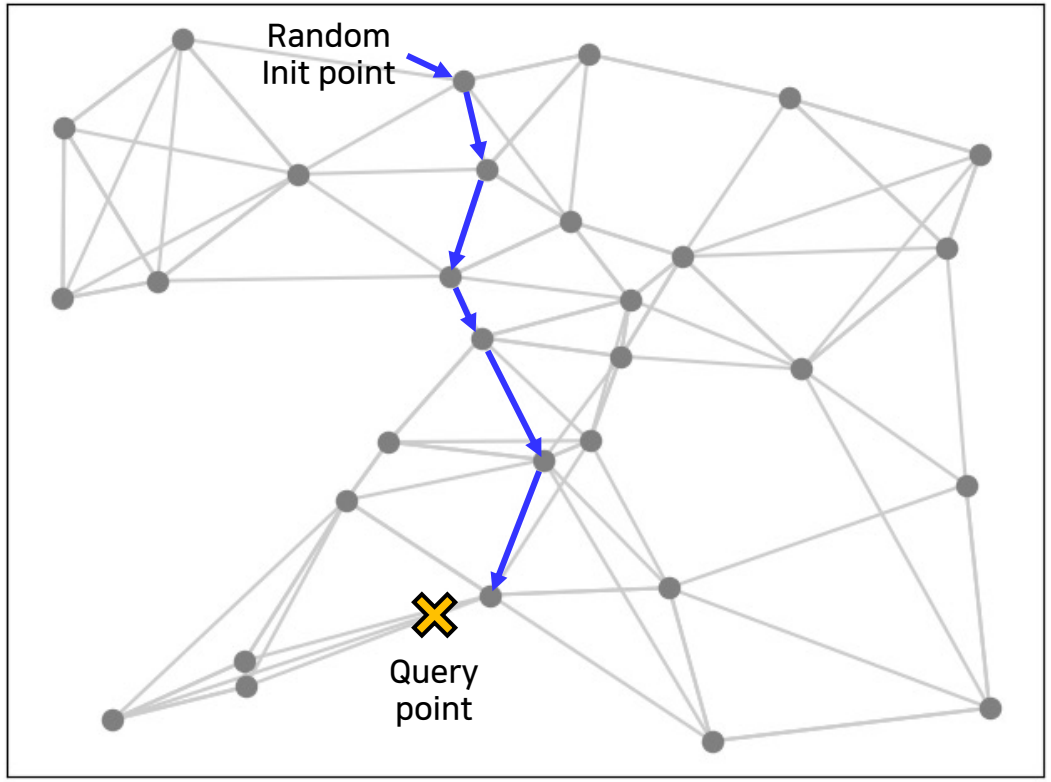
(Approximate Nearest Neighbor, ANN)

Intro : HNSW



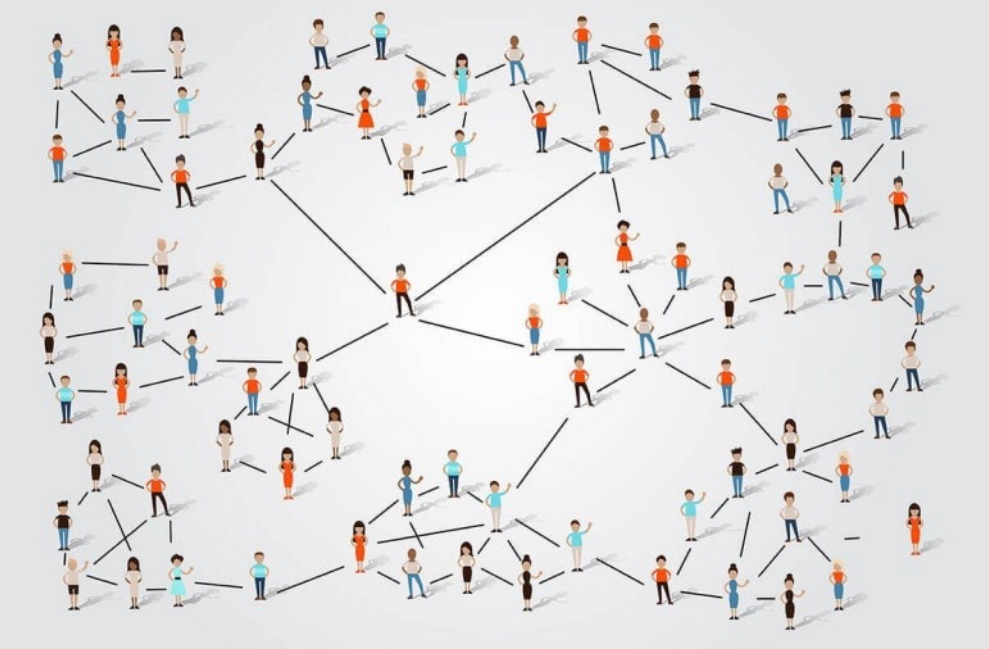
- HNSW
  - ANN 분야의 “근본” 방법 중 하나
    - 인용 수 1395회
  - 현재까지도 뛰어난 성능
    - ANN Benchmark 높은 순위
  - 자주 사용되는 ANN 중 하나

Background



- Graph-based KNN Search
  - 효과적인 KNN 탐색 방법 중 한 종류
  - 빌드 시, 가까운 점들 간 연결해 KNN 그래프를 생성해 둠
  - 쿼리 시, 쿼리 지점과 가까워지는 방향으로 점차 이동

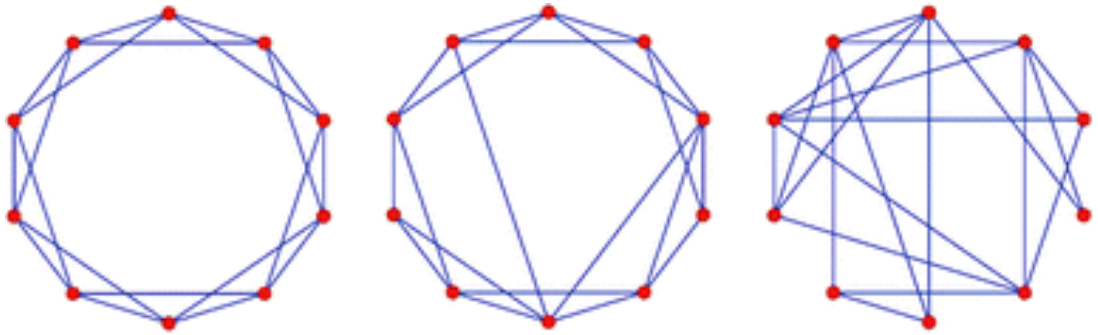
Background :Small world network



- 케빈 베이컨의 6단계 법칙
  - “지구상 대부분의 사람은 6단계만 거치면 연결 된다”
  - 세상 참 좁다 -> Small world network

Background :Small world network

랜덤성 P=0 ▸ P=1

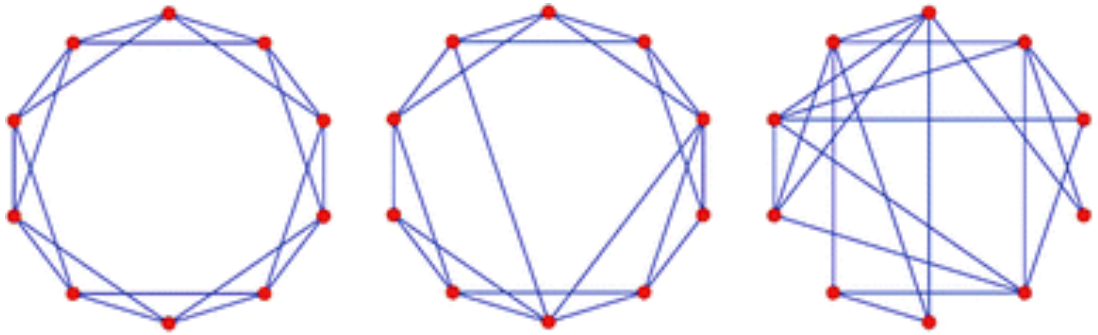


	KNN 그래프	Small world	랜덤
경로 길이	길음	**짧음**	짧음
클러스터링 계수	높음	**높음**	낮음

- Small world network
  - 임의의 두 점 간의 경로 길이가 짧음
    - 케빈 베이컨의 6단계 법칙
  - 클러스터링 계수가 높음
    - 유사한 아이템들끼리 잘 모여있음

Background :Small world network

랜덤성 P=0 ▸ P=1



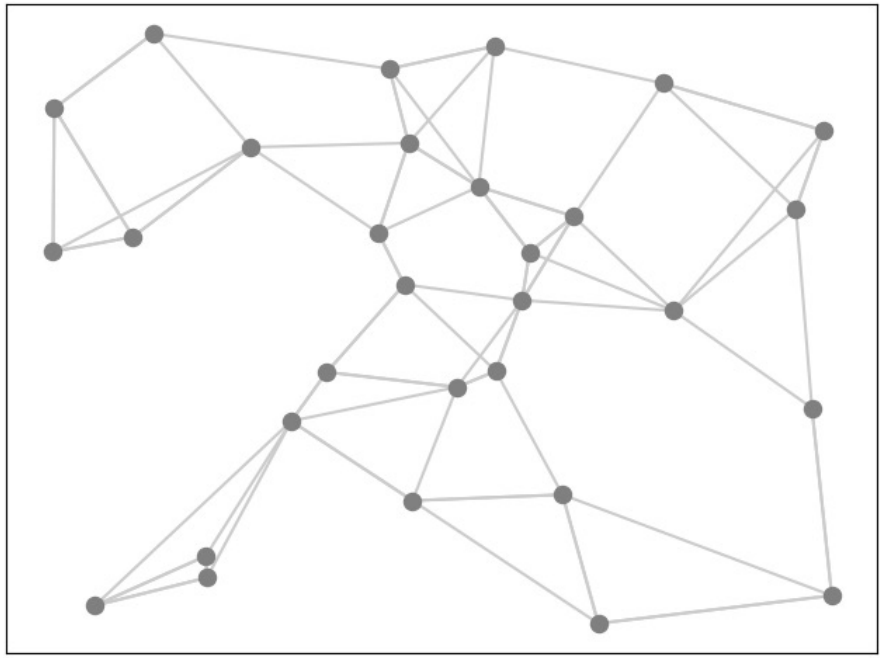
	KNN 그래프	Small world	랜덤
경로 길이	길음	**짧음**	짧음
클러스터링 계수	높음	**높음**	낮음

- Small world network
  - 임의의 두 점 간의 경로 길이가 짧음
    - 케빈 베이컨의 6단계 법칙
  - 클러스터링 계수가 높음
    - 유사한 아이템들끼리 잘 모여있음

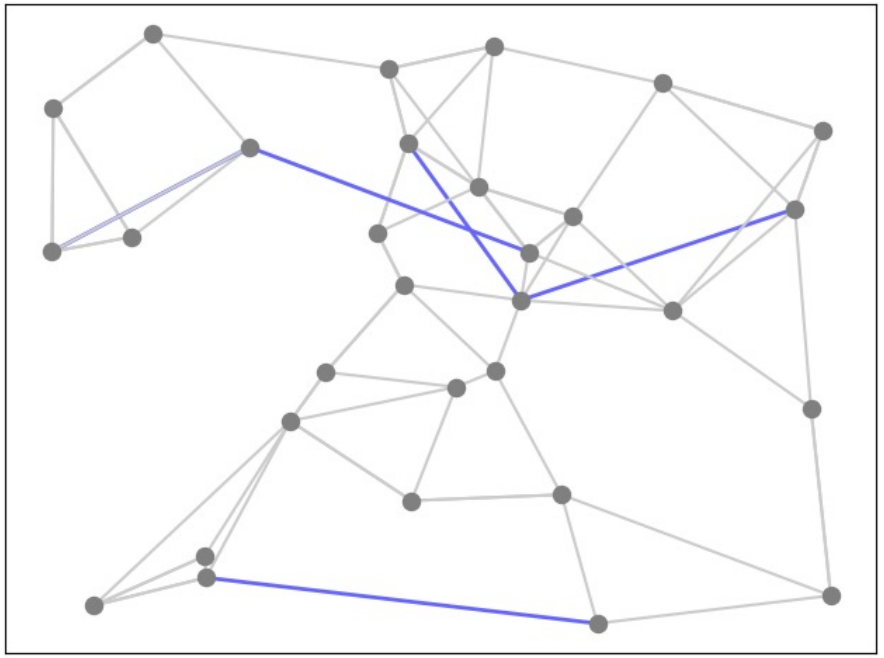
→ 그래프 기반 KNN 탐색에 적합!

(Navigable Small World, NSW)

Background :Navigable Small World

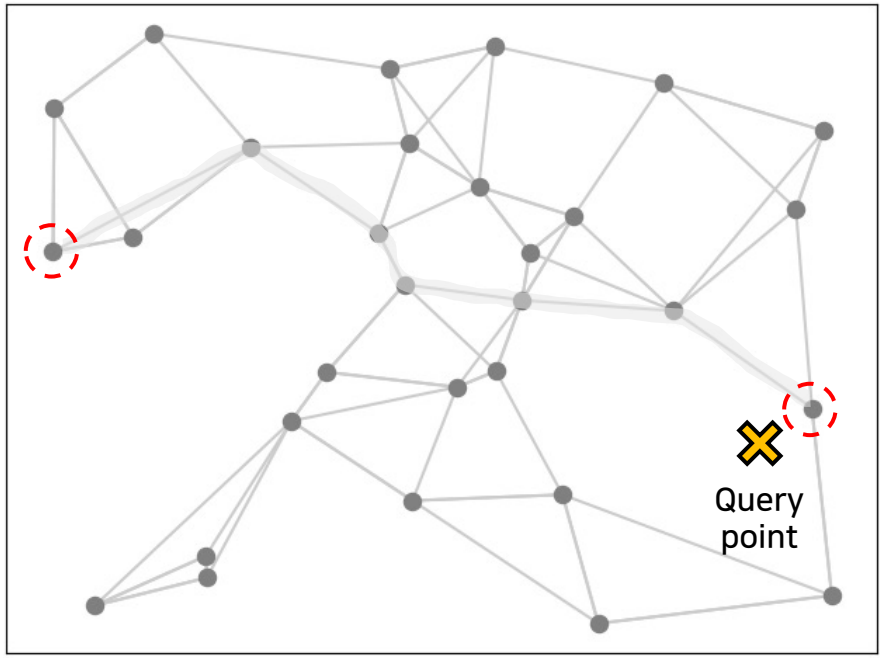


KNN 그래프



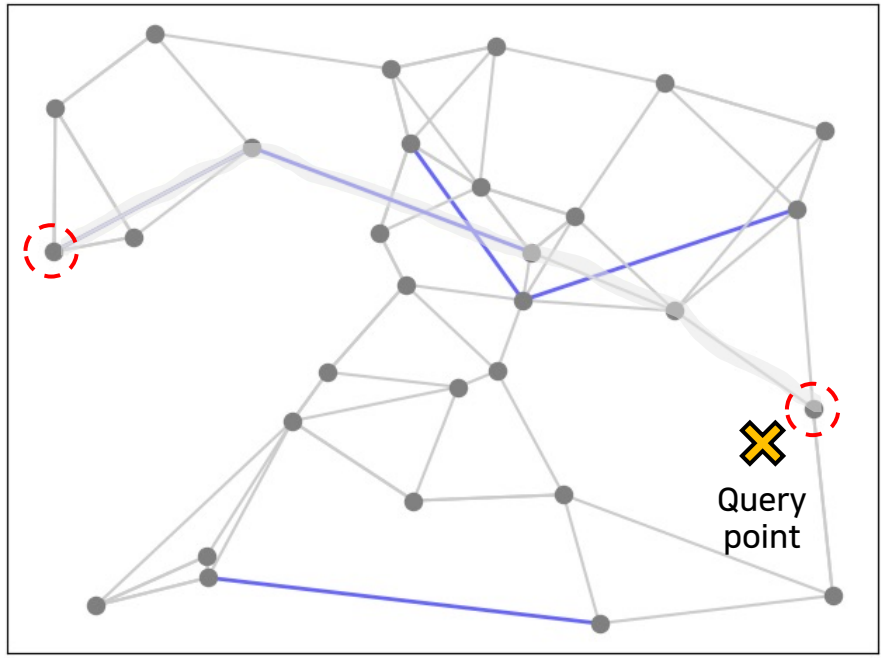
Small world 그래프

Background :Navigable Small World



KNN 그래프

경로 길이 6

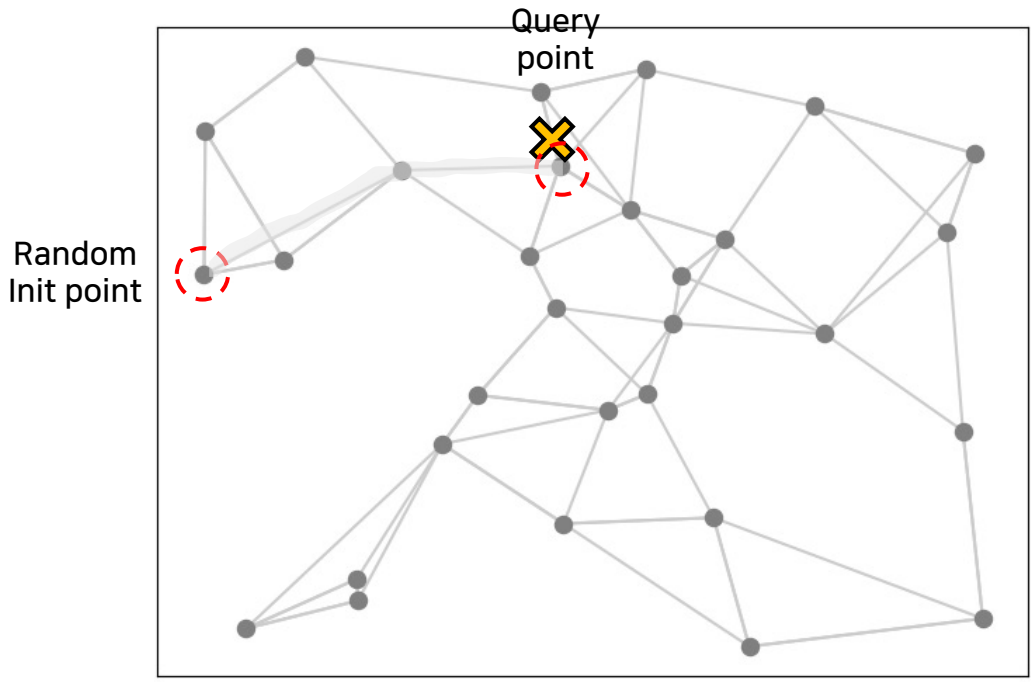


Small world 그래프

경로 길이 4

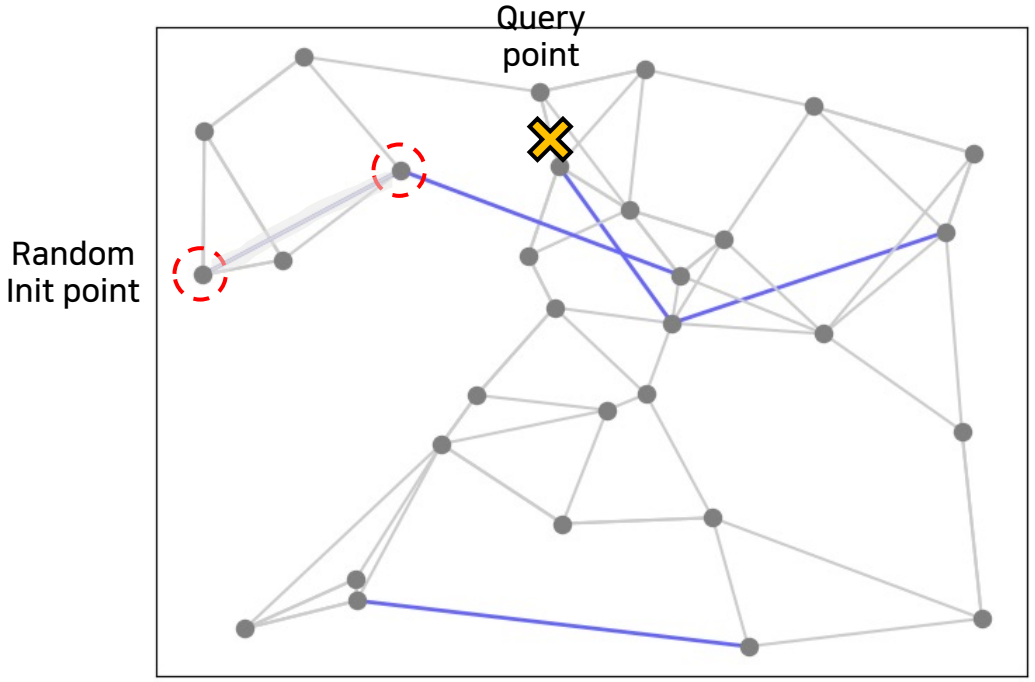
→ 더 빠르게 KNN 검색 가능!

Background :Navigable Small World



KNN 그래프

이웃을 잘 찾음

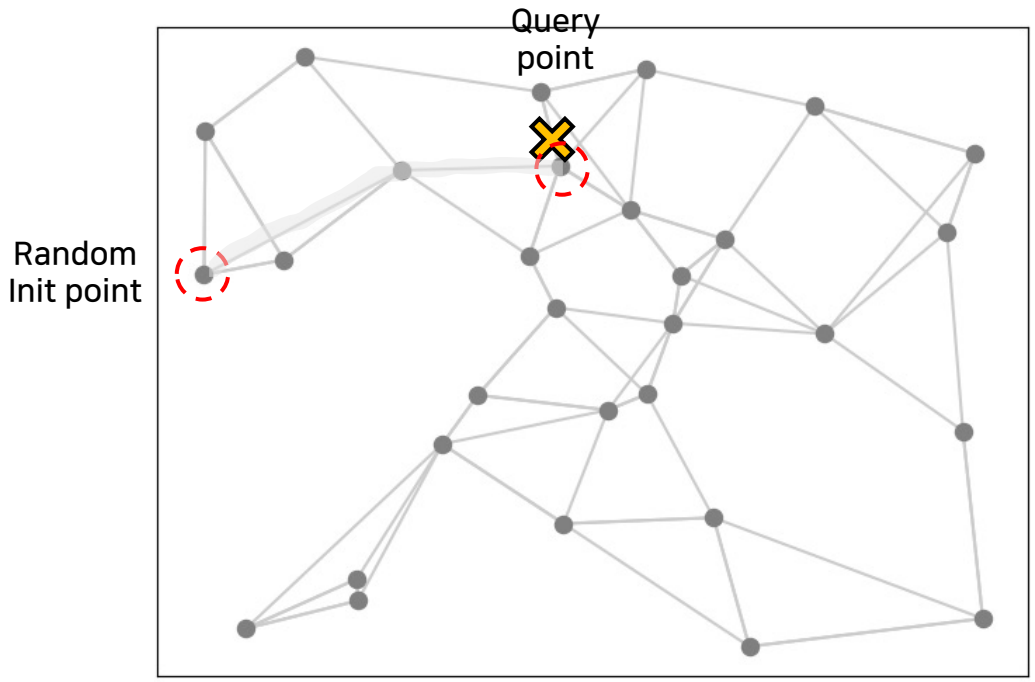


Small world 그래프

Local minimum에 빠짐

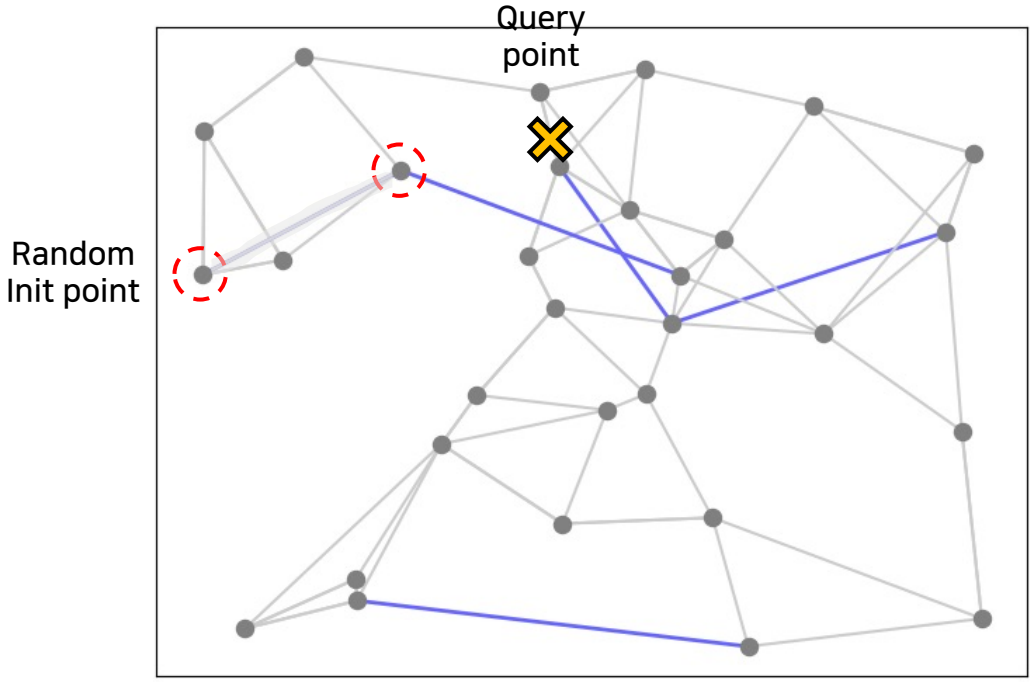
→ 정확도가 낮아질 수 있음..

Background :Navigable Small World



KNN 그래프

이웃을 잘 찾음



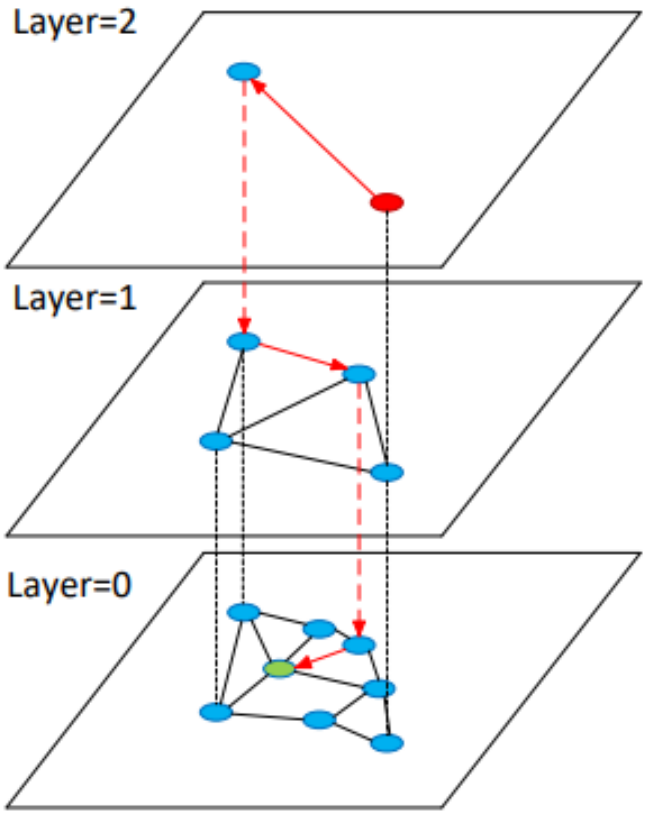
Small world 그래프

Local minimum에 빠짐

→ 정확도가 낮아질 수 있음..

→ Hierarchical Navigable Small World!

Method :HNSW Overview



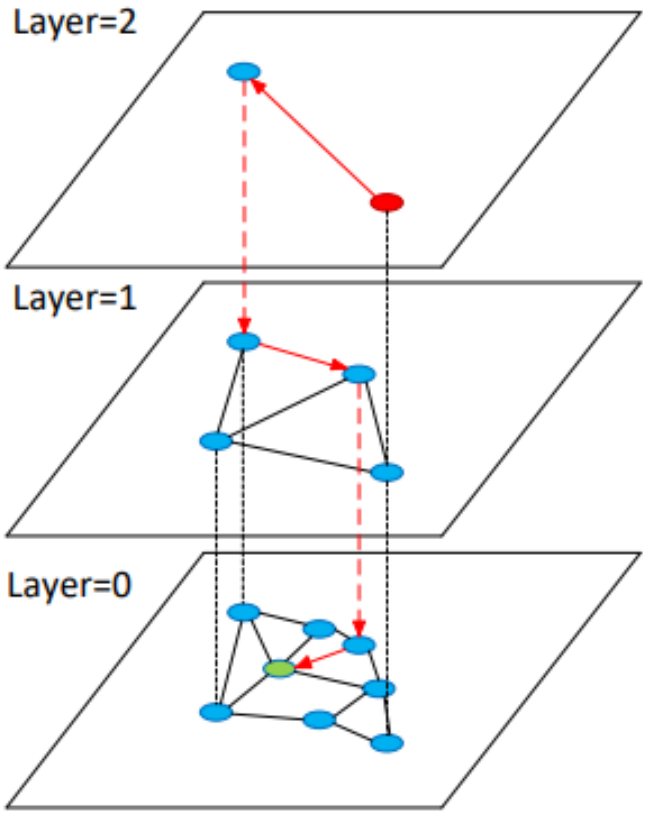
- **Hierarchical** Navigable Small World
  - 여러 개의 레이어로 나누어 구성
  - 확률적으로 레이어를 선택, 위로 갈수록 적은 수의 노드만 선택
  - 상위 레이어 노드인 경우 하위 레이어들에도 존재

Method :HNSW Query

검색(1개)만 수행

검색(1개)만 수행

K개 찾아 반환



- 쿼리 처리 방법
  - 가장 윗 레이어부터 검색 시작
  - "레이어 검색" 수행
    - 레이어 내에서 **그래프 KNN 검색** 수행하여 가장 가까운 점 찾기

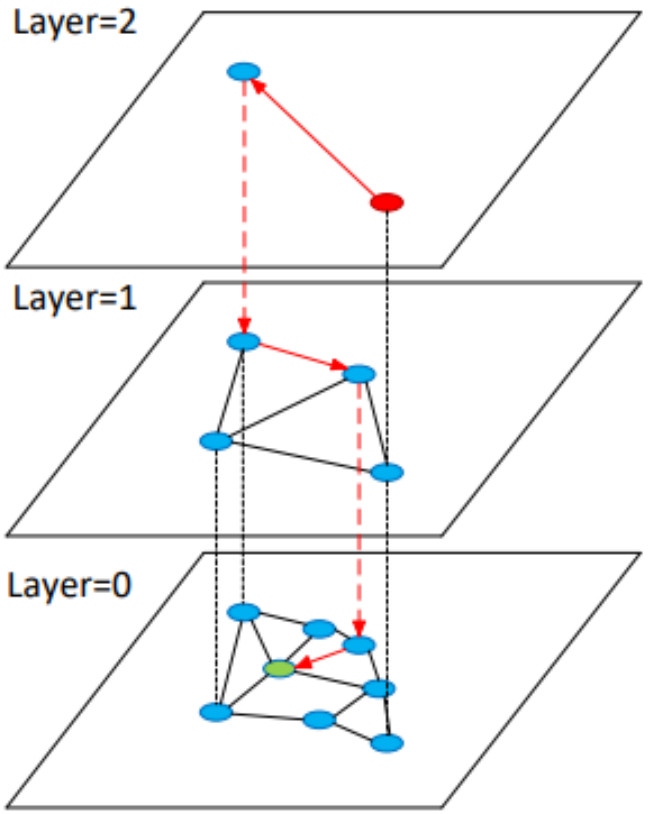
    (레이어 0인 경우에만 K개의 점 찾기, 상위 레이어에서는 1개의 점만 찾기)
  - 하위 레이어의 같은 노드로 이동해서 반복

Method :HNSW Query

검색(1개)만 수행

검색(1개)만 수행

K개 찾아 반환



- 쿼리 처리 방법
  - 가장 윗 레이어부터 검색 시작
  - "레이어 검색" 수행
    - 레이어 내에서 **그래프 KNN 검색** 수행하여 가장 가까운 점 찾기

    (레이어 0인 경우에만 K개의 점 찾기, 상위 레이어에서는 1개의 점만 찾기)
  - 하위 레이어의 같은 노드로 이동해서 반복

윗 레이어를 통해 빠르게 거리 줄이기 (NSW 장점 유지)

아래 레이어를 통해 Local minimum에 갇히지 않음 (NSW 단점 해결)

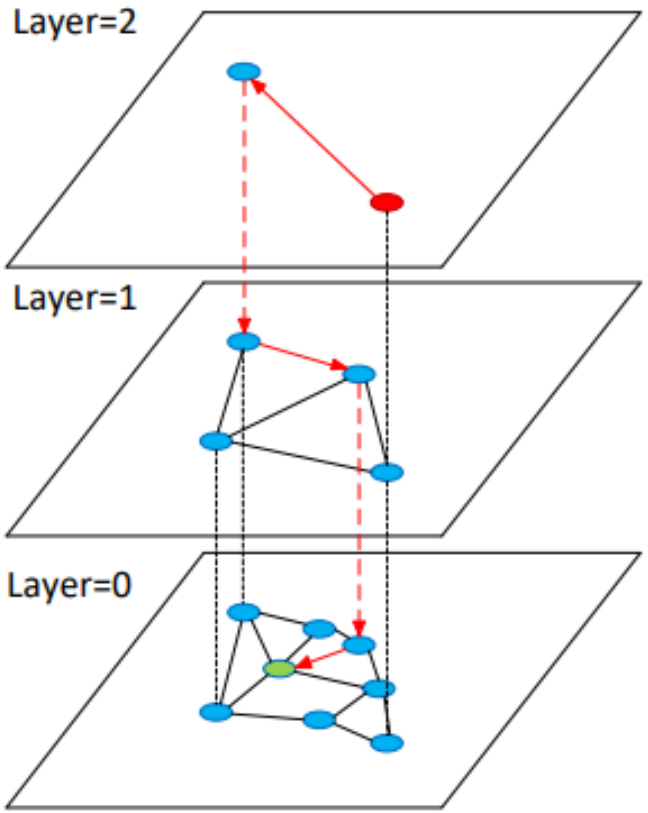
Method :HNSW Insert

Ex) l=1

검색(1개)만 수행

K_build개 찾아
그래프 갱신

K_build개 찾아
그래프 갱신

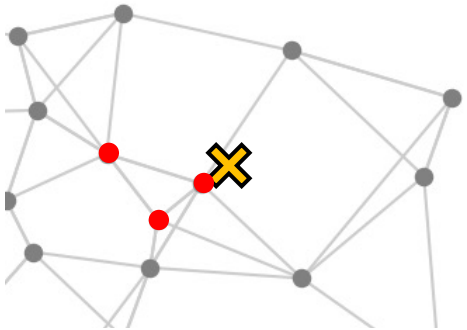


- 데이터 삽입 방법
  - 삽입될 노드의 목표 레이어 l을 정함
  - 삽입될 노드를 쿼리 대상으로 하여 **HNSW 쿼리**를 수행

    (레이어 l 이하인 경우에만 k_build 개의 점 찾기, 상위 레이어에서는 1개의 점만 찾기)
  - l 이하의 레이어에서는 찾은 k_build 개의 점과 연결해 그래프 갱신
  - 하위 레이어에서도 반복

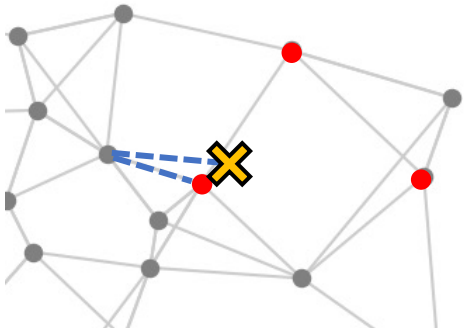
Method :HNSW Insert (휴리스틱)

k_build = 3



단순 이웃 선택

가장 가까운 3개



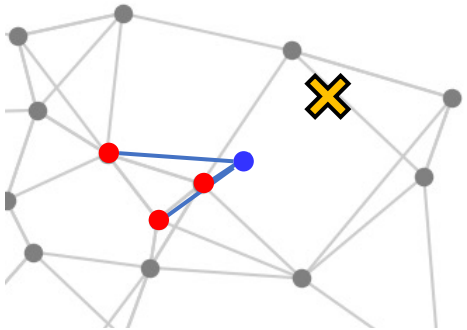
휴리스틱 이웃 선택

다양한 방향 3개

- 휴리스틱 이웃 선택 방법
  - 이웃을 더 다양한 방향으로 연결해주기
  - 가까운 순서대로 연결
  - 연결하려는 점 기준,
    삽입 점보다 가까운 이미 연결된 점이 있다면 연결하지 않음
  - 이미 이 방향으로 연결된 더 가까운 점이 있는 경우를 회피

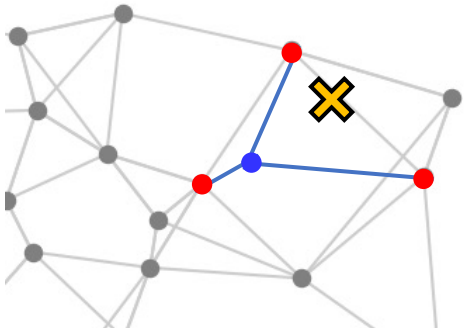
Method :HNSW Insert (휴리스틱)

k_build = 3



단순 이웃 선택

가장 가까운 3개



휴리스틱 이웃 선택

다양한 방향 3개

- 휴리스틱 이웃 선택 방법
  - 이웃을 더 다양한 방향으로 연결해주기
  - 가까운 순서대로 연결
  - 연결하려는 점 기준,
    삽입 점보다 가까운 이미 연결된 점이 있다면 연결하지 않음
  - 이미 이 방향으로 연결된 더 가까운 점이 있는 경우를 회피

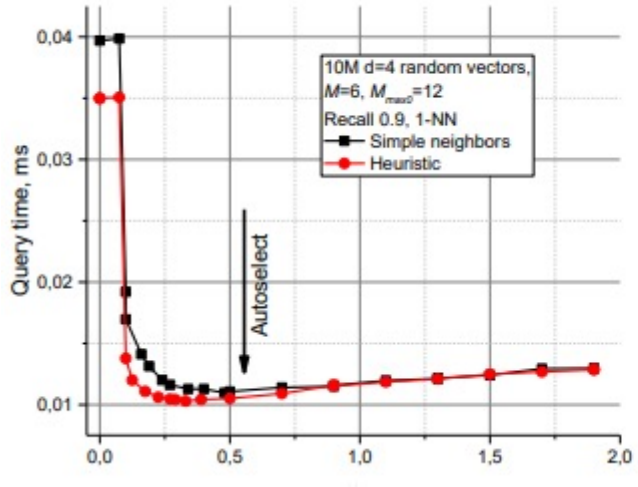
다양한 방향으로 연결하여 Local minimum에 갇히지 않게 함

탐색 정확도 향상

Experiments :파라미터 실험

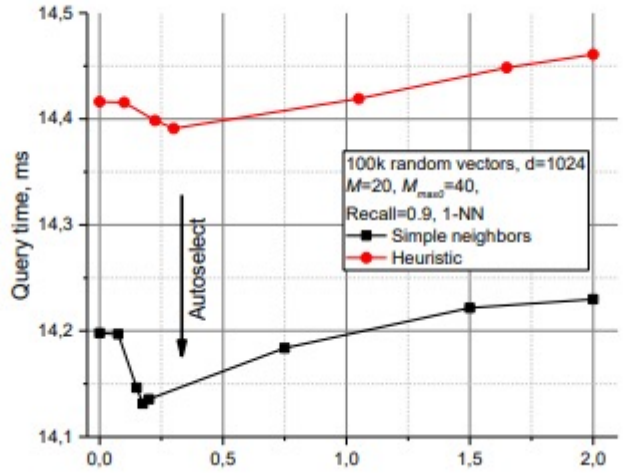
X축: 레이어의 분포를 결정하는 값, 높을 수록 많은 레이어

Y축: 탐색에 걸리는 시간



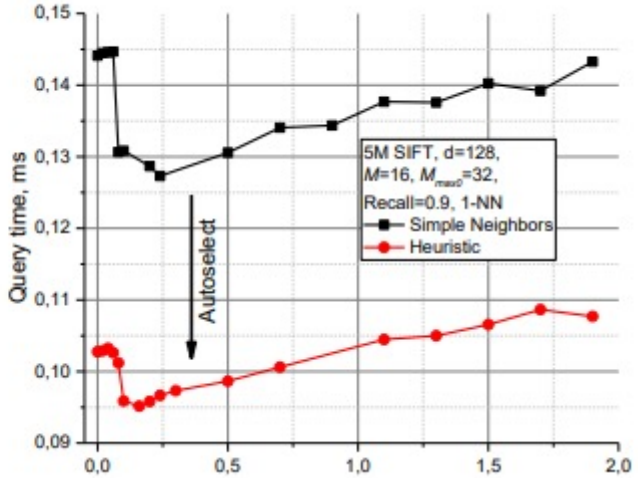
랜덤 데이터

10M, d=4



랜덤 데이터

100k, d=1024



실제 데이터 (SIFT)

5M, d=128

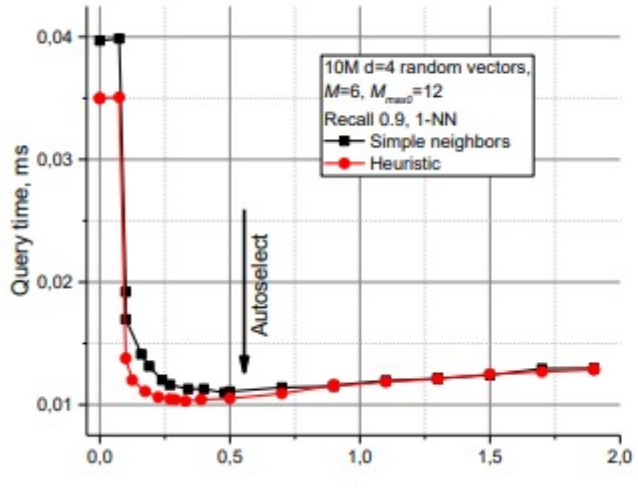
휴리스틱 방법(빨간색)이 실제 데이터에서 더 뛰어남

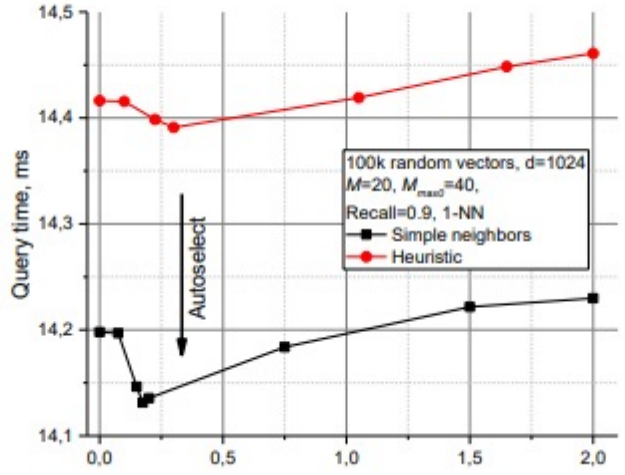
단, 차원이 아주 높은 경우에는 더 느려짐

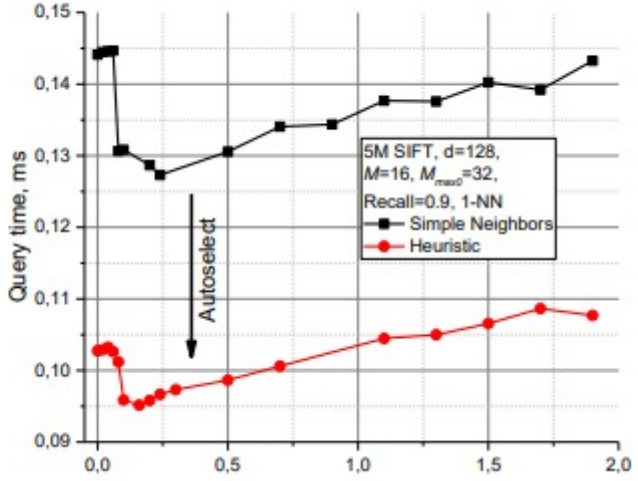
Experiments :파라미터 실험

X축: 레이어의 분포를 결정하는 값 mL, 높을 수록 많은 레이어

Y축: 탐색에 걸리는 시간







랜덤 데이터

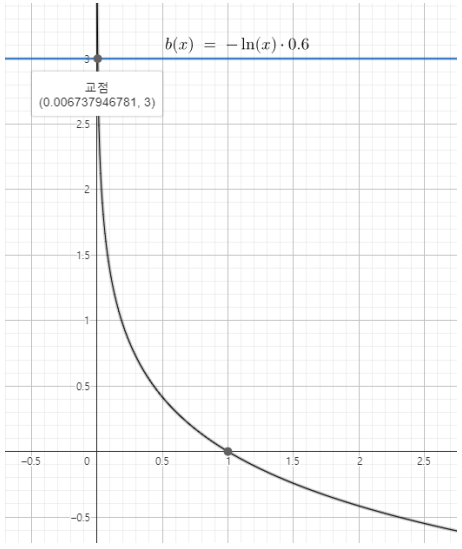
10M, d=4

랜덤 데이터

100k, d=1024

실제 데이터 (SIFT)

5M, d=128

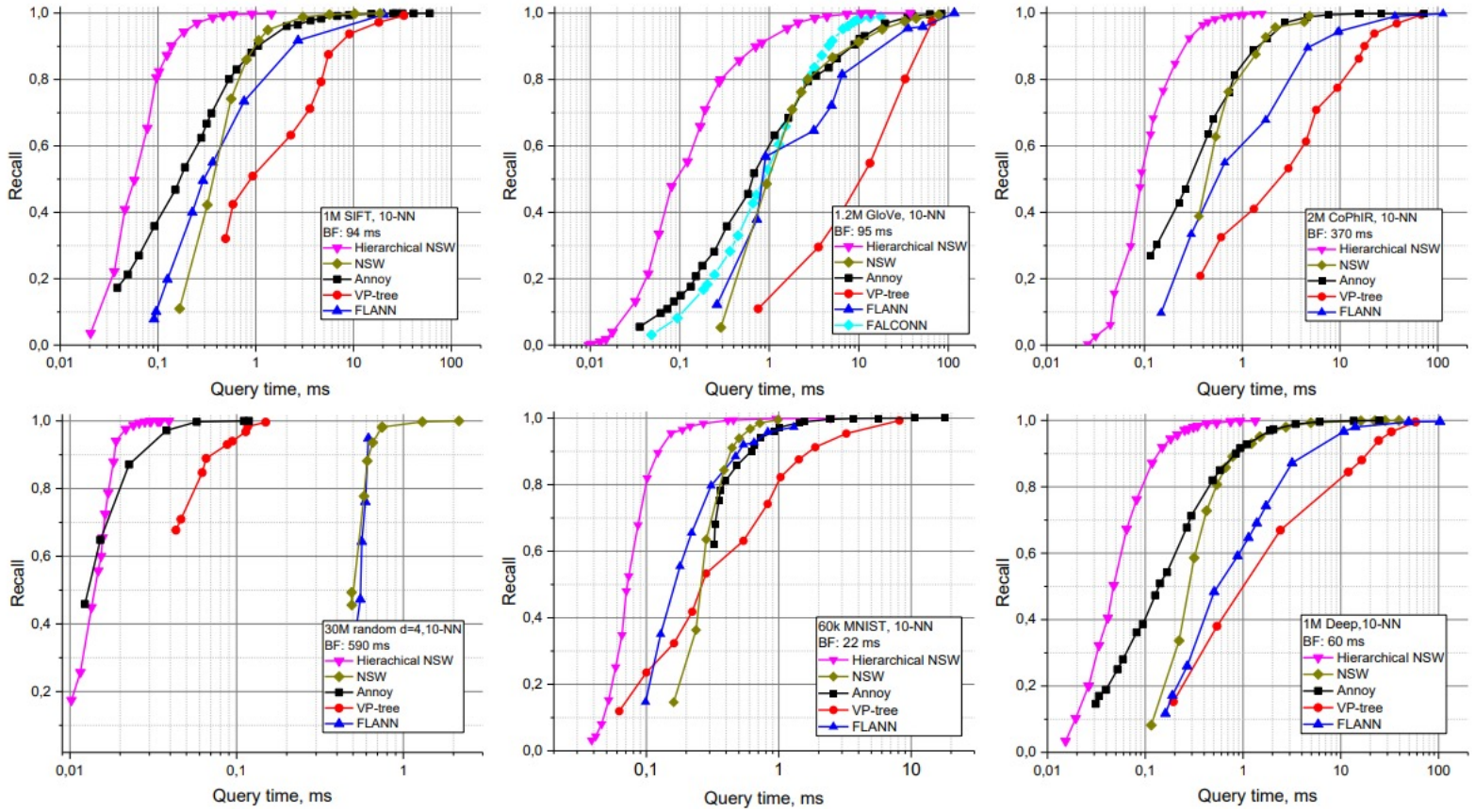


삽입될 레이어 결정: $-\ln(\text{unif}(0,1)) * mL$

mL이 높을 수록 분포의 최대값과 평균값이 상승

시뮬레이션을 통해 mL을 1 / ln(노드별 이웃 수) 로 선택하니 잘 되었음 (Autoselect)

Experiments :성능 실험



NSW 및 경쟁자들보다 모든 데이터셋에서 좋은 성능

Conclusion

- 계층 구조를 통해 NSW의 탐색 속도 강점을 가지면서도 높은 정확도도 확보함
  - 단순한 아이디어지만 굉장히 효과적
- 데이터 삽입이 가능
  - 삽입 시 다시 빌드를 해야 하는 알고리즘들이 많음
- 지정해야 하는 파라미터가 그래프의 이웃 수 하나 뿐 -> 파라미터 조절이 쉬움
  - 높은 성능을 보이기 위해 파라미터를 잘 조절해야 하는 경우가 흔함

아직까지도 ANN 알고리즘 중 상위권의 성능

추가자료