K-Nearest Neighbor Graph

Hirakata

July 22, 2015

読んだ論文

W. Dong, C. Moses, K. Li:
"Efficient K-Nearest Neighbor Graph Construction for Generic Similarity Measures" in *WWW 2011*

- `http://dl.acm.org/citation.cfm?id=1963487`
- `http://www.cs.princeton.edu/cass/papers/www11.pdf`

最 (k) 近傍探索問題

点集合 $V = \{v_1 \ldots v_n\}$ が与えられたとき,
全ての v_i に就いて, その最近傍 (あるいは k 位まで) の点を列挙せよ

全ての点対の距離を調べる: $O(n^2)$

k 近傍グラフ (kNN Graph)

点集合 V が与えられたとき

notation

点 $v(\in V)$ の k 位までの近傍点集合を $B_k(v)$ と書く

V を頂点集合とし, $v \to u$ $(u \in B_k(v))$ を有向枝とするグラフを kNN Graph という. 実質的には グラフは (V, B_k) で表現される

---

u を近隣とする点集合

$$R_k(u) = \{v : u \in B_k(v)\}$$

他の近傍点探索アルゴリズム

- tree-based data structure
  - kd 木
  - wavelet 木 とか
- Locality Sensitive Hashing

Efficient kNN Graph Construction: W.Dong, M.Charikar, K.Li

- kNN Graph はナイーブに作ると $O(n^2)$
- (経験的に) $O(n^{1.14})$ なアルゴリズムを紹介する
- 距離空間に依存しないことを目指す
- ただし、近似的なグラフ
- C++で実質的に 200 行以下

他諸々

Definition

距離空間 V に就いて,
点 $v \in V$ を中心とする半径 r の閉集合 (r 閉近傍) を

$$B_r(v) = \{u \in V : d(v, u) \leq r\}$$

と書く. ある定数 c を用いて

$$|B_{2r}(v)| \leq c \cdot |B_r(v)|$$

と書けるとき, V を growth-restricted metric space という
また, 上を満たす最小の c を growing constant という

基本のアイデア

- 厳密な kNN (V, B_k) ではなく近似グラフを構成する
- $B_k(v), R_k(v)$ の近似 $B[v], R[v]$
  - $U[v] := B[v] \cup R[v]$
- 近傍の近傍は近傍 (であり易い)
- 近似 $B[v]$ を逐次改善することを考える

N.B.

以降 $B_\cdot(\cdot)$ を r 閉近傍 (添字は距離) とし,
$B[\cdot]$ を近似の近傍点集合 (大きさは k) とする

近傍の近傍

適当な kNN Graph の近似 B が与えられたとき
点 v の近傍の近傍とは,

$$B'[v] = \bigcup_{v' \in B[v]} B[v']$$

- $|B[v]| = k$ とすると
- $|B'[v]| \leq k^2$

Prop

$B[v] \cup B'[v]$ から v の近傍 k 点を選ぶと, v との最長距離は一定の確率以上で半減する (後述)

アルゴリズム

procedure ApproximateKNNGraph(V, d)

$B[v]$ にランダムな k 点を割り当て
loop
  $B[]$ から $R[]$ を構成 (逆辺を張る)
  $U[] \leftarrow B[] \cup R[]$
  for $v \in V$ **do**
    for $v' \in U[v]$, $u \in U[v']$ **do**
      $B[v] \leftarrow \mathit{Update}(B[v], u)$
  Update される $B[v]$ がなくなったら終了

- ***Update*** は 1 点を追加して距離の小さい k 点に更新する操作
- アルゴリズム中で距離の計算 (d) をするのはここだけ
- $B[]$ を近傍点とその距離の組のコレクションだとし, ヒープ等で予め距離でソートしておけばこれは $O(\log n)$

距離が半減する確率

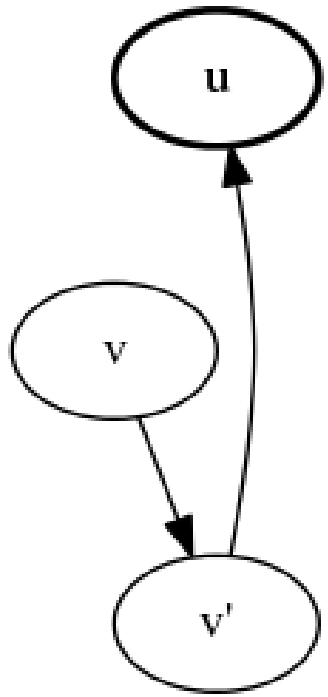
kNN Graph の近似 B の於ける v と $B[v]$ との最長距離を r とする

$$u \in B'[v] \iff \exists v' \, (v' \in B[v] \wedge u \in B[v'])$$

ある $u \in R_{r/2}(v)$, $v' \in R_{r/2}(v)$ に就いて
$v' \in B[v] \wedge u \in B[v']$ がなる確率を考える

- $Pr\{v' \in B[v]\} \geq k/|B_r(v)|$
- $Pr\{u \in B[v']\} \geq k/|B_r(v')|$

これは、$B[v]$ が $B_r(v)$ から一様に選ばれた集合であると仮定してる
2 式目は三角不等式から u が v から距離 r 以下にあることを使う



距離が半減する確率 cont

$v' \in B[v]$ と $u \in B[v']$ とが独立だとして

$$Pr\{v' \in B[v] \wedge u \in B[v']\} \geq \frac{k^2}{|B_r(v)||B_r(v')|}$$

点集合 V が 定数 c で growth restrected な距離空間だと仮定すると ($|B_r(v)| \leq c \cdot |B_{r/2}(v)|$)

$$Pr\{v' \in B[v] \wedge u \in B[v']\} \geq \frac{k^2}{c^3|B_{r/2}(v)|}$$

距離が半減する確率 cont

$Pr\{v' \in B[v] \wedge u \in B[v']\} \geq p$ という下限が見積もれた
更に $\exists v' \left(v' \in R_{r/2}(v)\right)$ を考えれば

$$Pr\{u \in B'[v]\} \geq 1 - (1-p)^{|B_{r/2}|}$$
$$\approx \frac{k^2}{c^3 |B_{r/2}(v)|}$$

$r \rightarrow$ 小, $B_{r/2}(v) \rightarrow$ 小, $Pr \rightarrow$ 大
(直感に反する?)

改良ポイント

- Local Join
- Incremental Search
- Sampling
- Early Termination

Local Join

2 重ループのメモリアクセスの局所性を増やす

v を固定したループ:

- for $v \in V$
  - for $v' \in U[v]$, $u \in U[v']$

だったのを

v' を固定した 2 重ループ:

- for $v' \in V$
  - for $v \in U[v']$, $u \in U[v']$

とするだけ

Incremental Search

更新した $U[]$ (近傍リスト) だけチェックする

2 重ループで前回のループで $U[v']$ が更新されてないとき

- for $v \in U[v']$, $u \in U[v']$

を飛ばす

Sampling

$|B[v]| = k$ であるが $|U[v]|$ に制限はない
ただし $\sum_v |U[v]| = 2\sum_v |B[v]| = 2nk$

3 重ループ

for $v' \in V$ **do**
  for $v \in U'$, $u \in U'$ **do**
    $\cdots$

のループ数 $(\sum_v |U[v]|^2)$ は

- 最良で $O(nk^2)$,
- 最悪で $O(n^2k^2)$.

Sampling

$|B[v]| = k$ であるが $|U[v]|$ に制限はない
ただし $\sum_v |U[v]| = 2\sum_v |B[v]| = 2nk$

3 重ループ

for $v' \in V$ **do**
  for $v \in U'$, $u \in U'$ **do**
    $\cdots$

のループ数 $(\sum_v |U[v]|^2)$ は

- 最良で $O(nk^2)$,
- 最悪で $O(n^2k^2)$.

そもそも「近隣の近隣」は重複しやすいため
全て調べるのは無駄

Sampling cont

$U[v']$ から ρK だけサンプリングして使う $(\rho \leq 1)$

3 重ループ

for $v' \in V$ **do**
  $U' \leftarrow SAMPLE(U[v'], \rho K)$
  for $v \in U'$, $u \in U'$ **do**
    ...

ループ数は $O(nk^2)$

Early Termination

全体の loop を完全に収束する前に, 早めに切り上げる

loop
  for $\cdots$ **do**
    $\cdots$
  if *Update* される点の数が δKN 以下 **then**
    return

計算量

以上の改良をした上で全体の計算量は

$$\text{全体のループ数} \times O(nk^2 \log n)$$

計算量

以上の改良をした上で全体の計算量は

$$\text{全体のループ数} \times O(nk^2 \log n)$$

で、ループ数は…?

計算量

以上の改良をした上で全体の計算量は

$$\text{全体のループ数} \times O(nk^2 \log n)$$

で、ループ数は…?

$\Rightarrow$ 実験で示します

実験

5 つの (現実の) データセットと種々の距離尺度を使う

1. Corel: Corel 画像データベース
2. Audio: DARPA TIMIT collection. 英語文を読み上げた音声
3. Shape: 3D モデル
4. DBLP: 書物に関するデータベース (テキスト). 著者の名前とか出版物のリストとか
5. Flicker: 画像

Dense vectors

1. Corel: 14 次元
2. Audio: 192 次元
3. Shape: 544 次元

のベクトルとして, それぞれ表す

距離に L_1, L_2 の 2 つを使う

- $L_p = \sum_i |\delta x_i|^p$

Text Data

1. DBLP

語のベクトル, 多重集合で表現して, 類似度に cosine と Jaccard を使う

- $cosine(x, y) = \frac{x \cdot y}{||x|| \cdot ||y||}$
- $Jaccard(x, y) = \frac{|x \cap y|}{|x \cup y|}$

類似度のマイナスを距離とする

Earth Mover's Distance

Flicker には EMD を使う

重み付きの素性ベクトル $\{\langle w_i, v_i \rangle\}_i$ $(\sum_i w_i = 1)$
どうしの距離を測る

画像 $\mapsto$ 領域への切り分け
$\mapsto$ 領域ごとの素性ベクトルと、領域の広さ (重み)

評価尺度: 近似の良さ, 計算効率

recall と scan rate を考える
各頂点について:

真の kNN Graph で正しく張られてる (枝数) / k

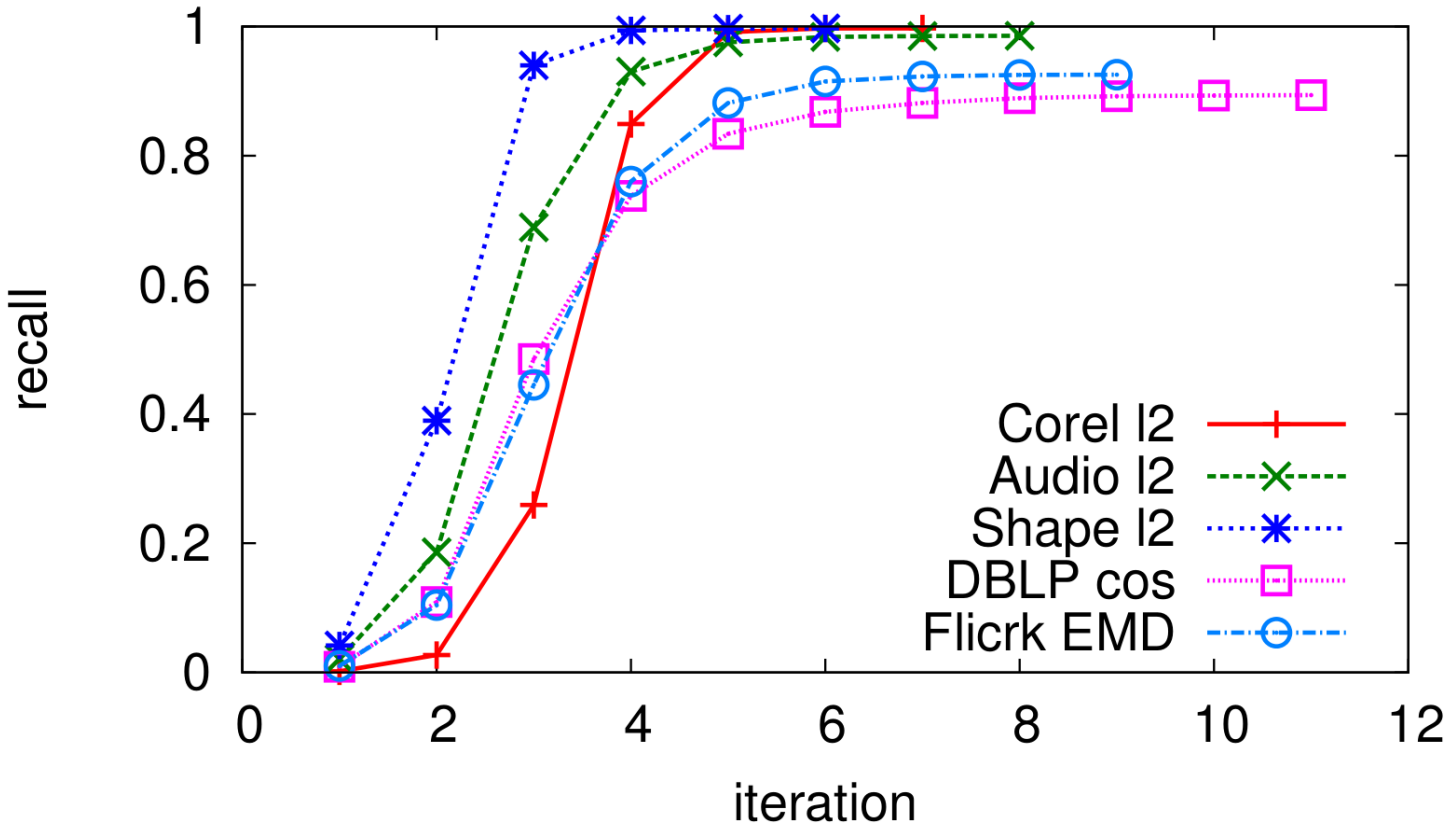
を計算して, 全頂点のそれの平均をグラフ全体の recall とする

ナイーブには距離の計算は $N(N-1)/2$ 回必要
実験で実際に行った計算の回数の割合を scan rate とする

パラメータ

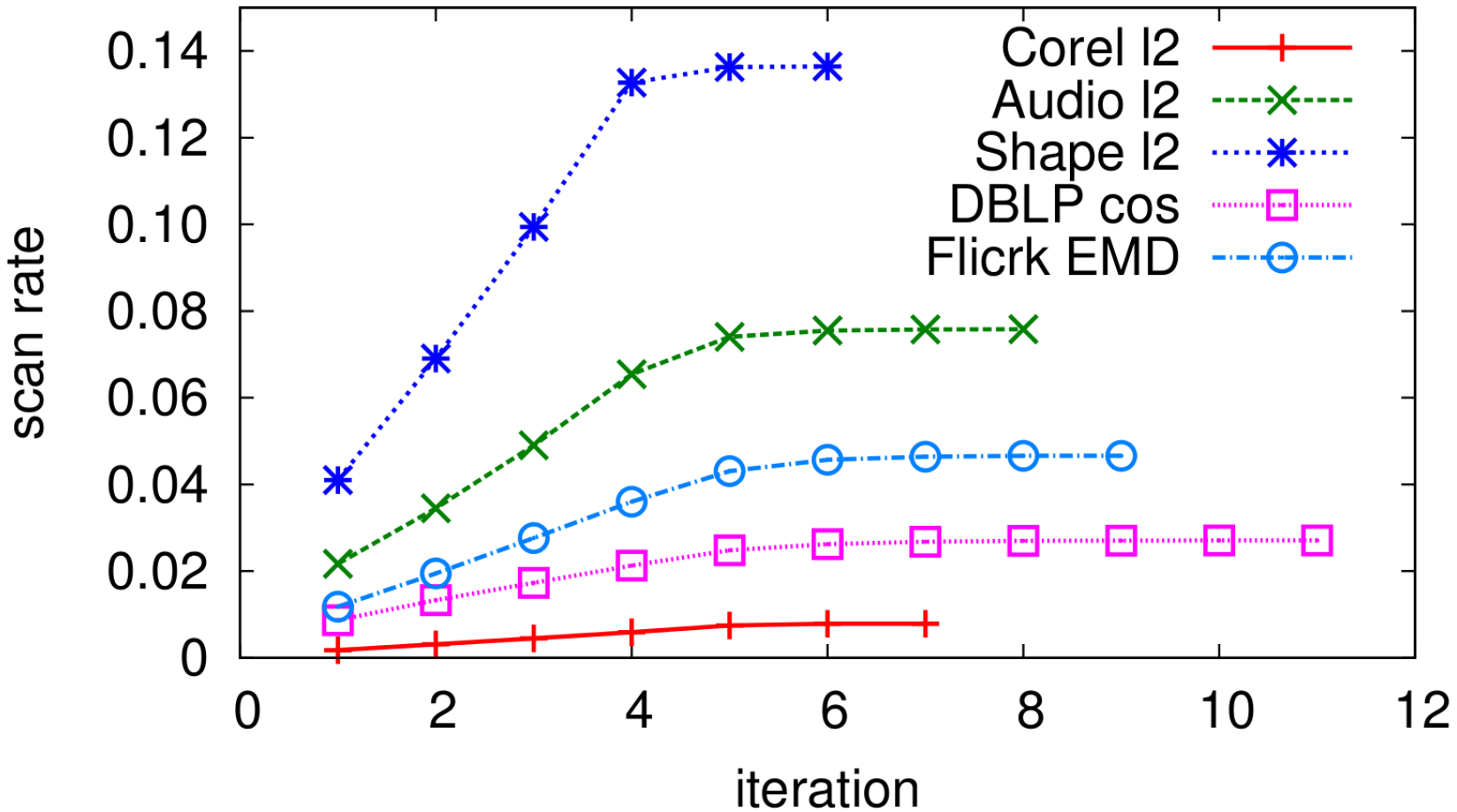
- $K = 20$
  - DBLP (テキスト) だけ $K = 50$
  - K は大きいほど有利
  - K を変更した場合の挙動もあとで見せます
- $\rho = 0.5, 1.0$
- $\delta = 0.001$

結果 – recall



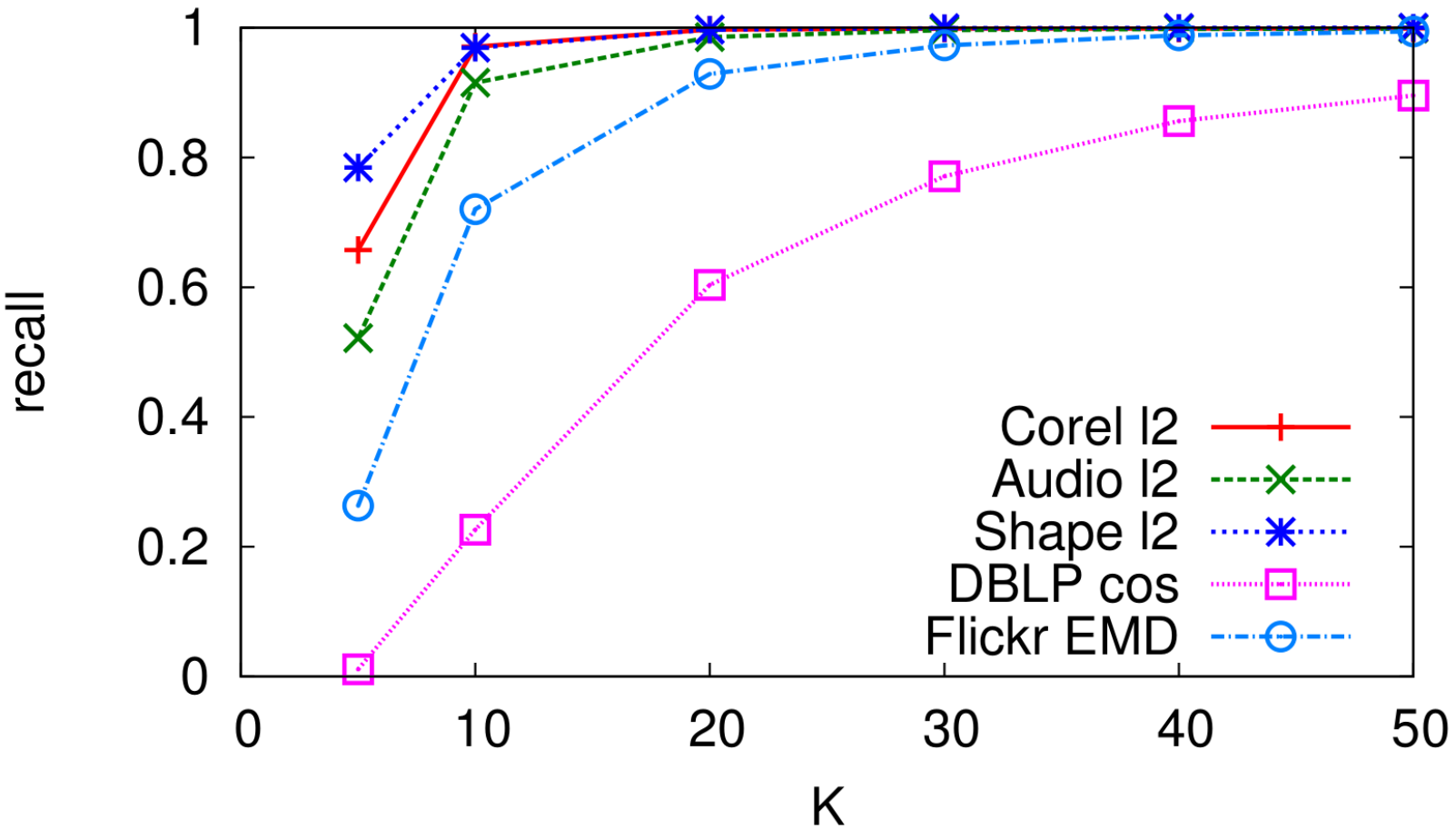
横軸はループ回数
全て 11 以下で終了している

結果 – rate scan

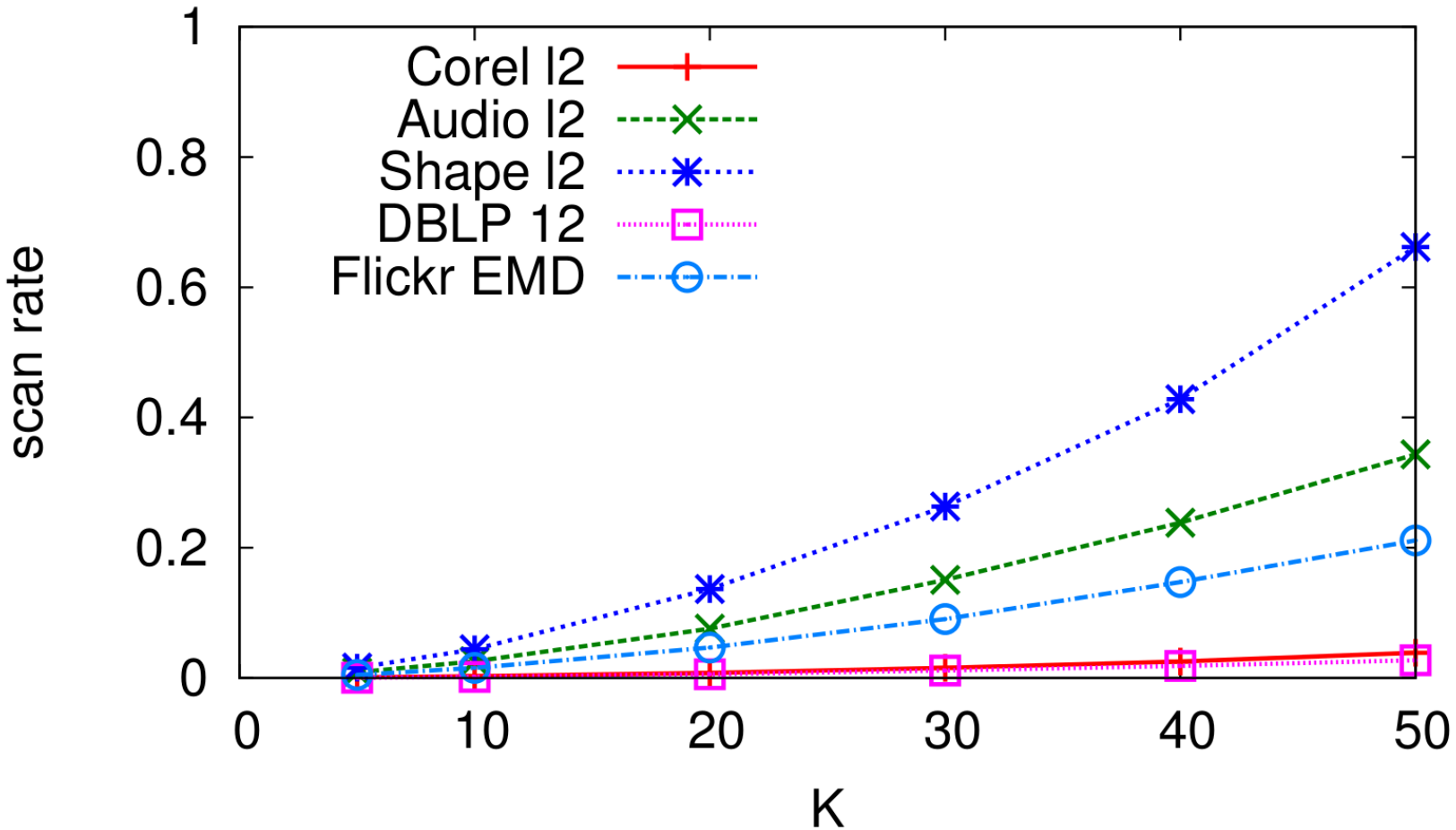


小さいほど良い

K を変えた時の挙動

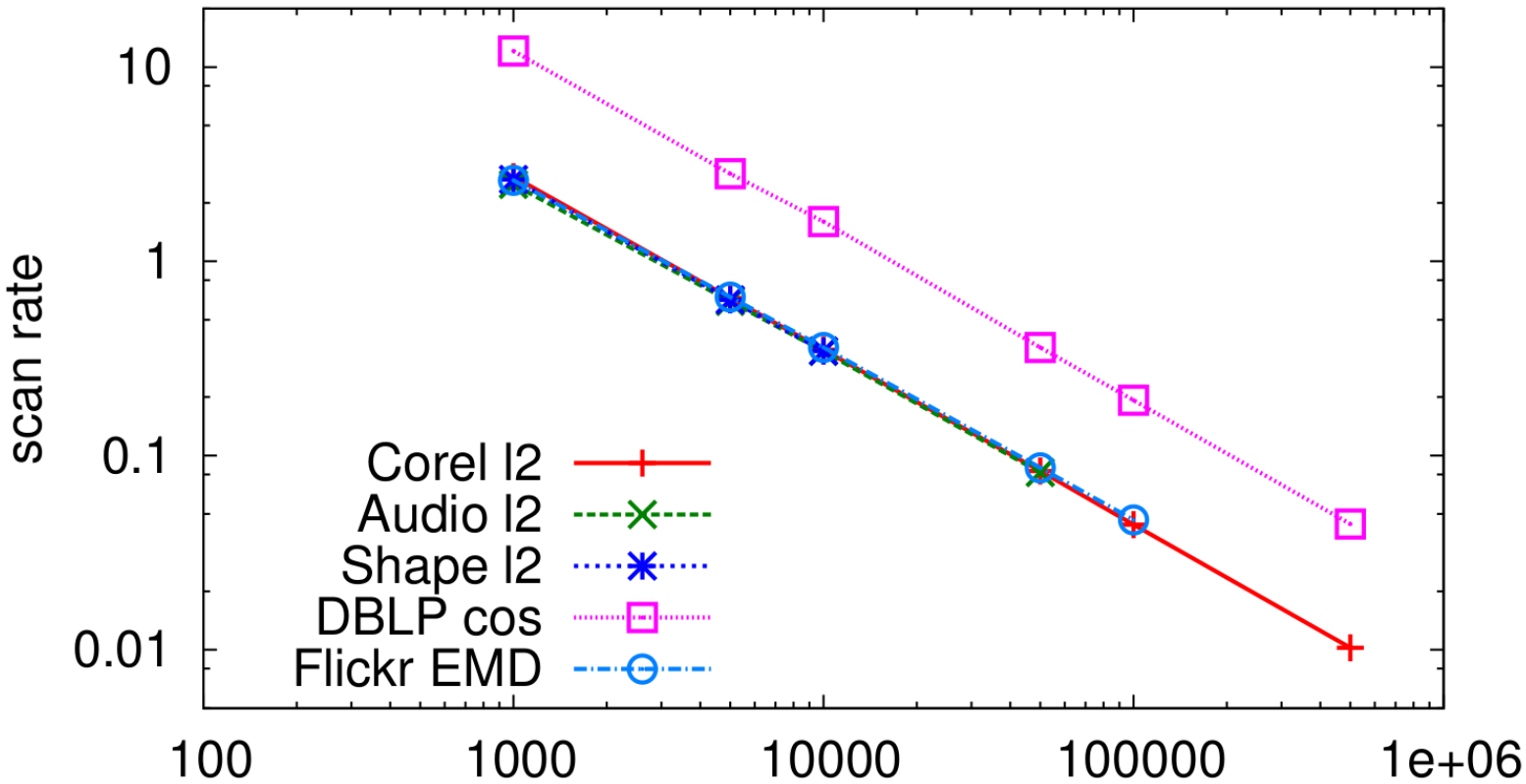


K は大きいほど recall は高くなる


scan rate
1
0.8
0.6
0.4
0.2
0
0
10
20
30
40
50
K
Corel l2
Audio l2
Shape l2
DBLP 12
Flickr EMD

計算量は K^2 に比例

データ数 vs scan rate



Empirical complexity

Dataset & Measure	Empirical Complexity
Corel/l_2	$O(n^{1.11})$
Audio/l_2	$O(n^{1.14})$
Shape/l_2	$O(n^{1.11})$
DBLP/cos	$O(n^{1.11})$
Flickr/EMD	$O(n^{1.14})$

まとめ

- kNN Graph の乱択による近似的な構成
- recall の scan rate のトレードオフ
- 全体計算量はデータセットによらず同じ傾向にあった