ERATO セミナ 2010 - No. 16

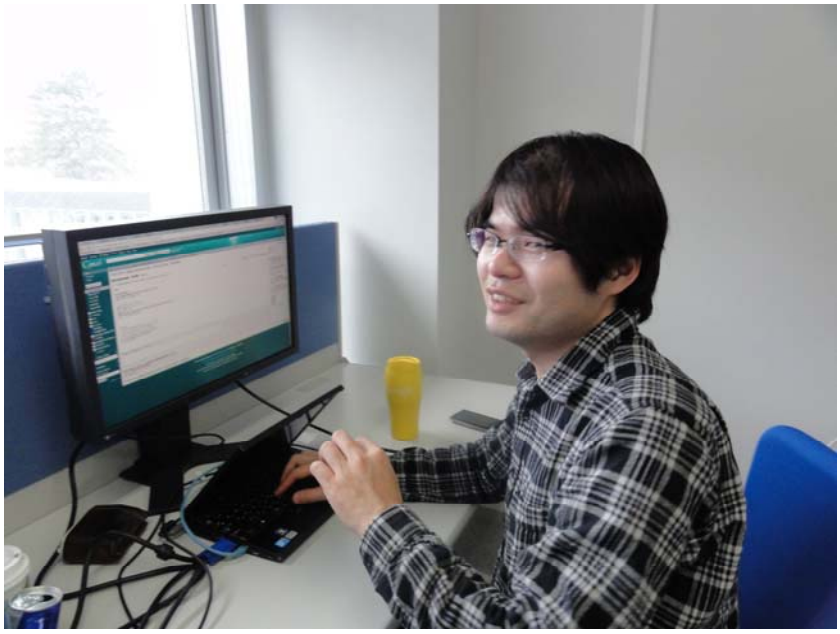
スケッチソート法による全点間類似度検索

田部井靖生
JST ERATO 湊離散構造処理系プロジェクト

2010/8/26

概要

近年、画像やテキストなどのウェブデーター、化合物やタンパク質などの生物学的データーなど、ベクトルとして表現されるデーターが大量に生成され、それらを効率的に処理する手法の開発が求められている。大規模データーから近傍検索を行う代表的な手法に Chariker (2002) らが提案した手法がある。これは、ベクトルデーターを 2 点間の距離関係を保ったままバイナリデーターへと射影した後、ソートとバイナリーサーチにより近傍点を検索する手法であるが、検索速度が遅いという欠点がある。そこで、我々は複合ソート法による全点間類似度検索法を提案する。提案手法は、ソート容量が大きくなる代わりに、近傍候補の距離計算を減少させることができ、高次元データーに対して効率的に類似度検索を行うことができる。大規模画像データーを用いた実験では、最近の近傍探索法との比較により提案手法の有効性が示を示す。



田部井研究員

ERATOセミナー@北大 2010/8/26

スケッチソート法による高速な全点間類似検索

ERATO湊プロジェクト研究員
田部井靖生
共同研究者:宇野毅明(NII), 杉山将(東工大), 津田宏治(産総研)

発表手順

1.動機
- 大規模データ解析法の必要性
- 全点間近傍検索

2.手法
- マルチプルソート法
- Cosine Locality Sensitive Hashing
- スケッチソート法

3. 実験
- 大規模画像データを用いた実験

4. 現在進行中のプロジェクト
- Jaccard係数への拡張
- 大規模化合物データへの適応

近年、大規模データーが大量に生成されている

- 80 million tiny images
  - 約8千万画像のデーター

http://groups.csail.mit.edu/vision/TinyImages/

- Pubchem

–約2千万(2010年8月現在)の化合物からなるデーター

http://pubchem.ncbi.nlm.nih.gov/

- NCBI Sequence Read Archive

–次世代シーケンサーから読まれた大規模配列データー

http://www.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?

など

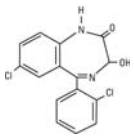
データーはベクトルとして表現できる

画像

sift特徴量

x_i=(0.2, −0.3, −1.3, 1.2, 2.2,...)

化合物



各次元は部分構造の有る(1)無し(0)を表す(フィンガープリント)

x_i=(1, 0, 1, 0, 1, …)

Localty Sensitive Hashing (Gionis et al. 99)

- ベクトルデーターを文字列に射影する方法
  - スケッチ:バイナリの文字列
  - 元の空間の２点間の距離関係を(ハミング距離で)保ったまま射影する
- 大規模データーを効率的に処理することが可能

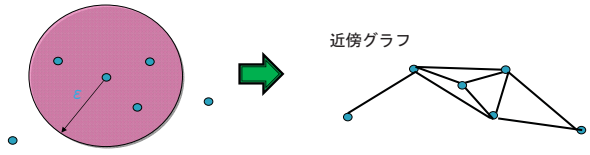
x=(0.2, −0.3, −1.3, 1.2, 2.2, …)

s=10101011101010101…

全点間類似度検索

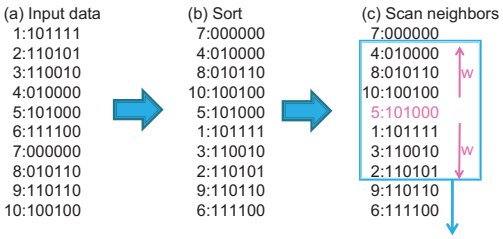
- データーセットからすべての似ているペアーを探す
- データーのペア(i, j), i < j, Δ(xi,xj) ≦ ε,

- 近傍グラフ(neighborhood graph)を作ることができる
- 応用
  - 重複データの削除, 大規模クラスタリング、半教師あり学習、画像から特徴的な領域発見、など



シングルソート法 (Charikar, 02)

- スケッチから全点間類似度検索を行う方法
- ソートしてある上下幅wのウィンドウでスキャンする
- 同じウィンドウに入ったペアーに関して、ハミング距離を計算する



シングルソート法の欠点

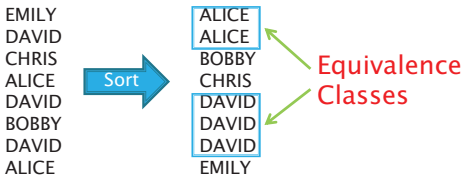
- 精度を上げるために多くの距離計算が必要
- 真の近傍ペアーを見逃す割合(fase negative)の理論的な見積もりができない

マルチプルソート法の概要(Uno, 08)

- 同じ長さlの文字列集合 $S=\{s_1,\ldots,s_n\}$を入力としてハミング距離d以内のペアの全列挙
- 距離d以内のペアの数をmとする
- 基数ソートを繰り返して、全ペアーをO(n+m)で列挙可能
- ブロックマスクを導入することにより高速化する

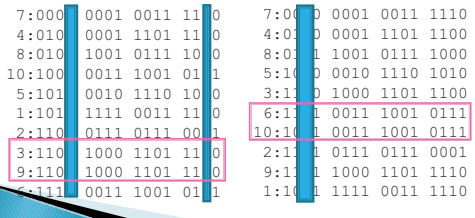
Special case: 同一文字列ペアの発見 (d=0)

- 基数ソートによりソートする、文字列を同一の文字列クラスに分割する:O(n)
- 各同一文字列クラスのすべてのペアーに対してエッジを張る:O(m)
- 計算量:O(n+m)



マルチプルソート法 (d>0のとき)

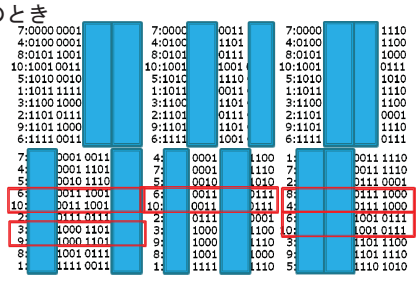
- d個の文字を全通り選んでマスクする
- 基数ソートを $\binom{l}{d}$ 回繰り返す
- 計算量はdに対して指数、lに対して多項式
- 文字列の数に対しては線形のまま O(n+m)



ブロックマスク

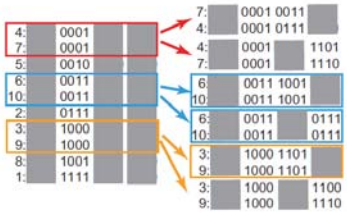
- 文字列をk個のブロックに分割する
- d個のブロックを、全通りマスクする
- ブロック単位なので、ソート回数が減る
- 近傍でないペアも検出されてしまう
  - 実際に検出されたペアのハミング距離を計算して排除する

k=4, d=2のとき



再帰的アルゴリズム

- 初めに第一ブロックをソートし、Equivalence classを発見する
- それぞれの、Equivalence classに対して、次のブロックを付け加えてソートする
- k-dのブロックをソートしたら、各Equivalence Classに入っているペアーに対して重複除去を行う
- 生き残ったペアに対して、ハミング距離を実際に計算



重複除去

- ブロックの組み合わせに対して辞書順をつける

1	2	3	4

- k=4, d=2のとき
- (1,2)<(1,3)<(1,4)<(2,3)<(2,4)<(3,4)
- あるブロックの組み合わせに対して、完全一致するペアが見つかったら、それが最小の完全一致するブロックの組み合わせかをチェックする

1	2	3	4

- (2,4)が完全一致したら、(1,2),(1,3),(1,4),(2,3)が完全一致するかをチェックする
- 実際には，1と3だけをチェックすればよい

疑似コード

```
1: function MultipleSortingMethod
2:     I ← {1, ..., n}
3:     B ← φ
4:     Recursion(I, B)
5:     return
6: end function
7: function Recursion(I, B)
8:     if |B| = k − d then
9:         for (i, j) ∈ I × I, i < j do
10:            if s_i^b ≠ s_j^b for all b < max(B), b ∉ B then
11:                if HamDist(s_i, s_j) ≤ d then
12:                    Report (i, j) to output file
13:                end if
14:            end if
15:        end for
16:        return
17:    end if
18:    for b in (max(B) + 1), (k + |B| + 1) do
19:        J ← Sorted indices based on b-th block {s_i^b}_{i∈I}
20:        T ← Intervals of equivalence classes in {s_i^b}_{i∈I}
21:        for each interval (x, y) ∈ T do
22:            Recursion(J[x : y], B ∪ b)
23:        end for
24:    end for
25:    return
26: end function
```

K-dコの完全一致するブロックを持つペアーに対する重複チェックとハミング距離の計算

選ばれたブロックを基数ソートしてequivalence classに分割する

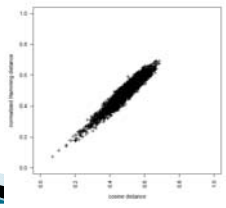
Equivalence classの各メンバーを再帰処理する

スケッチソートの概要

- 距離が閾値以内のすべてのペアーを列挙する
  (i, j), i < j, Δ(xi,xj) ≦ ε ,
- 基本的なアイディア
  - ベクトルデーターをLSHで文字列データーに射影する
  - 文字列データーをマルチプルソート法にて近傍ペアを列挙する
- LSHを変えることで、様々な距離尺度に対応可能
  -ユーグリッド距離 (Raginsky et al. 2010),コサイン距離(Geomans et al,1999), Jaccard係数(Broder,1998)など
- ここでは、コサインLSHを使った方法を紹介

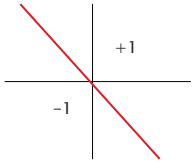
コサインLSH

- D次元のベクトルを長さlの0/1文字列に射影する
  - コサイン距離 $\Delta(x_i, x_j) = 1 - \frac{x_i^T x_j}{\| x_i \| \| x_j \|}$
- 射影先の正規化ハミング距離が、元の空間でのコサイン距離に単調増加する
  - Ex) 512bit



コサインLSH

- $R \in \Re^{D \times \ell}$:N(0,1)からサンプルされたランダム行列
- 写像 $s_{ik} := sign(r_k^T x_i), r_k \in R, k = 1,...,l$



- 非衝突確率は２点間の角度に比例

$$\Pr(s_{ik} \neq s_{jk}) = \frac{\theta_{ij}}{\pi}, \forall k,$$

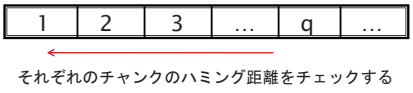
$$\theta_{ij} = \arccos\left(\frac{x_i^T x_j}{\| x_i \| \| x_j \|}\right)$$

スケッチソート

- Not good:長い文字列に、マルチプルソート法を適用する
- 文字列をチャンクへ分割
  - 長さℓの文字列プールをQ個つくる
  - マルチプルソート法を各プールに適用する
    $E_q = \{(i,j) \mid HamDist(s_i^q, s_j^q) \le d, i < j\}.$
  - 各プールにおける出力セットE_qの和集合をとる
    $E = E_1 \cup \cdots \cup E_Q.$
  - Eの中で、$\Delta(x_i, x_j) \le \epsilon$ を満たすペアを出力

チャンク単位の重複除去

- 異なるチャンクで同じペアが見つかると、重複したペアを出力してしまう
- 重複除去:チャンクqで、ハミング距離d以内のペアが見つかった場合は、チャンク1,…,q-1でハミング距離d以内のものがない場合だけ出力

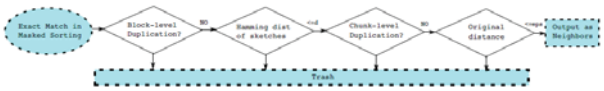


それぞれのチャンクのハミング距離をチェックする

各ペアに対して計３回のチェックを行う

1.ブロック単位の重複除去
2.ハミング距離の計算
3.チャンク単位の重複除去

- これらすべてを通過したペアのみコサイン距離を計算する



スケッチソートの疑似コード

```
1: function SketchSort(x_1, ..., x_n)
2:     Use LSH to obtain sketches {s_11, ..., s_qL}_{i=1}^n from
       data {x_i}_{i=1}^n
3:     I ← {1, ..., n}
4:     for q = 1 : Q do
5:         B ← φ
6:         Recursion(I, B, q)
7:     end for
8:     return
9: end function
10: function Recursion(I, B, q)
11:     if |B| = k − d then
12:         for (i, j) ∈ I × I, i < j do
13:             if s_iq ≠ s_jq for all b < max(B), b ∉ B then
14:                 if HamDist(s_i1, s_j1) ≤ d for all r < q
                       then
15:                     if Δ(x_i, x_j) ≤ ε then
16:                         Report (i, j) to output file
17:                     end if
18:                 end if
19:             end if
20:         end if
21:     end if
22:     end for
23:     return
24: end if
25: for b in (max(B) + 1)..(k + |B| + 1) do
26:     J ← Sorted indices based on b-th block {s_iq^b}_{i∈I}
27:     T ← Intervals of equivalence classes in {s_iq^b}_{i∈I}
28:     for each interval (x, y) ∈ T do
29:         Recursion(J[x : y], B ∪ b, q)
30:     end for
31: end for
32: return
33: end function
```

- 各チャンクに対する呼び出し
- Equivalence classの各メンバーのペアーを発見
- ３回のチェック
- Equivalence class に対しる再帰呼び出し

２種類のエラー

- 真のエッジセット E*, 中間結果 E
- Type-I error (false positive)： 近傍でないペアが、１つ以上のチャンクでハミング距離d以内となる事象
  $F_1 = \{(i,j) \mid (i,j) \in E, (i,j) \notin E^*\}.$
- Type Ⅱ-error (false negative)： 近傍ペアが、全てのチャンクでdより大きいハミング距離となる事象
  $F_2 = \{(i,j) \mid (i,j) \notin E, (i,j) \in E^*\}.$

Type Ⅱ-errorの上限: Missing edge ratio

- false negativeの方が、致命的
  - false positiveは、コサイン距離の計算により除去することができる
- Missing edge ratio(false negative rate)は次のようにバウンドすることができる

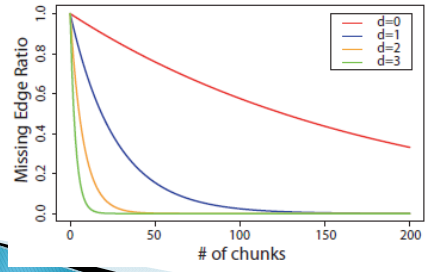
$$E\left[\frac{|F_2|}{|E^*|}\right] \le \left(1 - \sum_{k=0}^{\lfloor d \rfloor} \binom{\ell}{k} p^k (1-p)^{\ell-k}\right)^Q,$$

  - pは、LSHの非衝突確率の上限

$$p = \frac{\arccos(1-\epsilon)}{\pi}.$$

チャンクの数Qに対する、Missing edge ratioの減少

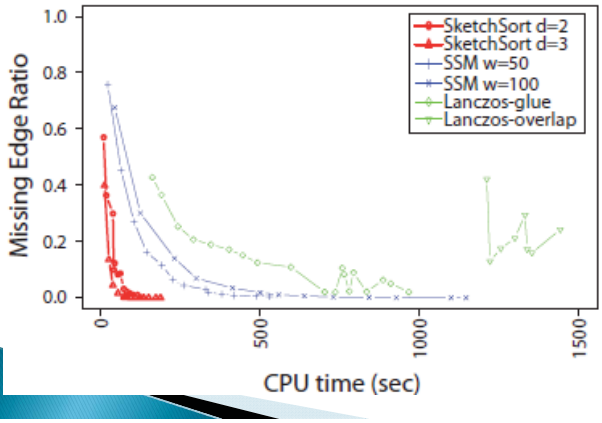
- チャンクの完全一致を用いる方法(d=0)では、十分にMissing edge ratioを減らせない



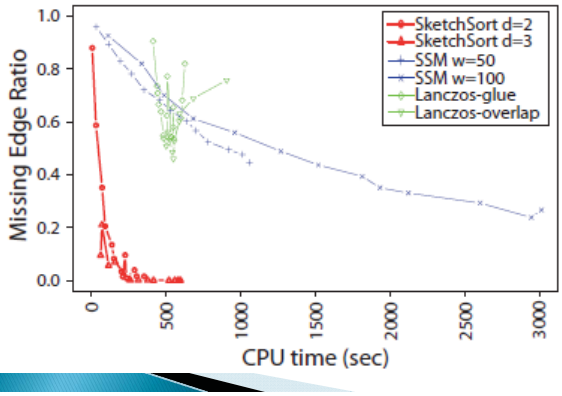
シングルソート法、Lanczos Bisectionとの比較

- データーセット
  - MNIST (60,000 データ 748 次元)
  - TinyImage (100,000 データ, 960 次元)
- 評価指標はコサイン距離 0.15π でのMissing egde ratio
- 各チャンクは32ビット
- マルチプルソート法のハミング距離とブロック数: (2,5)と(3,6)に固定
- チャンク数, 2, 6, 10,...,50
- 最新の近傍探索法Lanczos Bisection (JMLR,09)との比較
  - データのマージン最大を最大化する超平面で分割することを再帰的に繰り返し、分割領域内の点でナイーブ探索する

MNIST 閾値0.15π



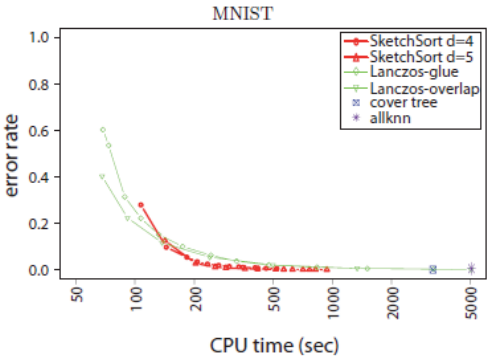
Tiny Image 閾値0.15π



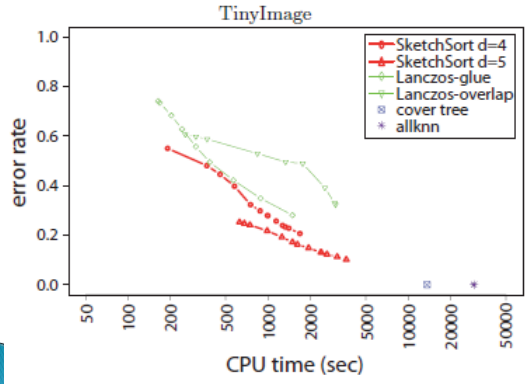
K-nearest neighbor検索実験

- スケッチソートの出力の各点ペアの距離が近いbest-h個をpriority queueで保持
- 全ての処理が終了した後、近傍グラフ上の各点から距離２以内のノードから、もっとも近い点を選ぶ(リファインメント)
- 比較手法
  - Cover Tree (Beygelzimer et al., ICML 2006)
  - AllKNN (Ram et al., NIPS 2009),
  - Lanczos-bisection (JMLR, 2009)

5-Nearest Neighbor検索 (MNIST)

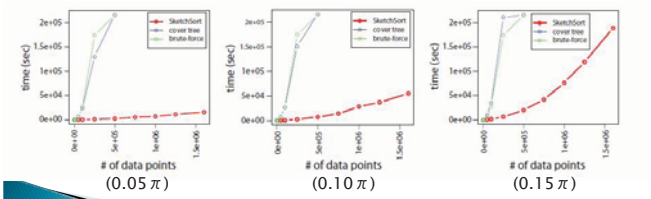


5-Nearest Neighbor検索 (TinyImage)



160万画像での実験

- False negative rate $< 1.0 \times 10^{-6}$
- 160万画像の全ペア類似度検索が、4.3時間 (0.05π)



(0.05π) (0.10π) (0.15π)

化合物の大規模解析

- Pubchem
  - 約2千万の化合物のFinger p rint
- 100万データを使ってKristensen et al.(Algo. for Mol. Biol., 2010) の手法と比較
- 距離尺度としてJaccard係数を使用
- ２つの集合X_i,X_jのJaccard係数

$$r(X_i, X_j) = \frac{X_i \cap X_j}{X_i \cup X_j}$$

- $1 - r(X_i, X_j)$で距離として利用

Minwise- independent permutations

- X⊂[n]に対する射影min(π_k(X))
  - π_k: [n]のランダム置換

n=10のとき

1	2	3	4	5	6	7	8	9	10

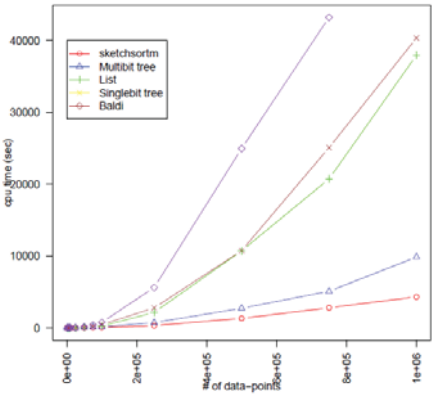
π

9	5	10	3	6	1	4	7	8	2

- このときXの各要素は等確率で最小要素に射影される(Minwise-independent)
  x∈X, Pr(min{π_k(X)}= π_k(x))=1/|X|
- 衝突確率はJaccard係数に比例する

$$\Pr(\min\{\pi(X_i)\} = \min\{\pi(X_j)\}) = r(X_i, X_j)$$

100万化合物, 閾値0.1
Missing edge ratio 1×10^{-6}以下



100万データで
約１時間

まとめ

- 高速な全点間類似度検索法を提案
- 大規模なベクトルデータに対して適応可能
- いろいろな応用を考えることができる
- コード
  http://code.google.com/p/sketchsort/