Embedded Stringology

Piotr Indyk

MIT

Combinatorial Pattern Matching

- Stringology [Galil] : algorithms for strings (as well as trees and other plants)
  - Classic/standard stringology: exact
    - String matching, suffix trees etc
    - Tools: automata theory, combinatorics on words
  - Non-standard stringology: approximate/noisy
    - Pattern matching with mismatches
    - Dictionary problems
    - Tool: FFT

Plan the talk

- Overview of problems
- Embeddings: what, why ?
- Embeddings for stringology
- Open problems

Noisy Pattern Matching

- Real life data is often noisy
- Algorithms should be robust to noise
- How to define noise ?
- Typically, via a distance function. E.g., when searching for pattern P, we accept substrings S such that $D(P,S) \leq k$

Distance functions

- Hamming: D(P,S)=H(P,S) = # indices i s.t. $P_i \neq S_i$
  - Simple and general
  - Not realistic ?
  - [Buhler, RECOMB'01] :

	A	G	T	C
A	+1	-1	-2	-2
G	-1	+1	-2	-2
T	-2	-2	+1	-1
C	-2	-2	-1	+1

→

	A	G	T	C
A	0	2	3	3
G	2	0	3	3
T	3	3	0	2
C	3	3	2	0

→

A	0AA
G	0GG
T	1TT
C	1CC

Distance functions ctd.

- L_p norms:
  - P_i and S_i are real numbers
  - $D(P,S)=||P-S||_p$

Distance functions ctd.

- Edit distance: D(P,S)=minimum number of operations needed to transform P to S
  - Typical operations:
    - Insertions, deletions, substitutions of characters (ED)
    - Swaps, etc.
    - Copies/reversals of whole blocks (BED)
  - Operations reversible $\Rightarrow$ D(P,S)=D(S,P)

Problems

- Pattern matching:
  - Exact: given T, $|T|=n$, and P, $|P|=m$, find substring S of T such that $D(S,P) \leq k$ (if it exists)
  - Approximate: can output a substring S' such that $D(S',P) \leq k(1+\varepsilon)$ (if a "$\leq k$ -match" exists)
- Near neighbor/dictionary/post-office problem:
  - Given $S = S_1 \ldots S_N$, $|S_i| \leq m$, build a data structure which does the following:
    - Given P, $|P| \leq m$, report S_i such that $D(S_i,P) \leq k(1+\varepsilon)$ (if a "$\leq k$ match" exists)
  - Variant: $S_1 \ldots S_N$ are all m-substrings of a text T

Problems Recap

- Pattern matching or near neighbor
- Under Hamming, L_p or Edit distances

Embeddings

Embeddings: Definition

- Assume we have $M_1=(X_1,D_1)$, $M_2=(X_2,D_2)$
- A mapping $f:X_1 \rightarrow X_2$ is a c-embedding if for any p,q from X_1 we have

$$D_1(p,q) \leq D_2(f(p),f(q)) \leq c*D_1(p,q)$$

- Example:

	A	G	T	C
A	0	2	3	3
G	2	0	3	3
T	3	3	0	2
C	3	3	2	0

A	0AA
G	0GG
T	1TT
C	1CC

Embeddings for Algorithms

Hamming metric

- Noisy pattern matching:
  - Exact:
    - $O(n\ |\Sigma|\ \log n)$ [Fisher-Paterson'74]
    - $O(nk)$ [Landau-Vishkin, Galil-Giancarlo'85]
    - $O\tilde{}(n\ m^{1/2})$ [Abrahamson, Kosaraju'89]
    - $O\tilde{}(n\ k^{1/2})$ [Amir-Lewenstein-Porat, SODA'00]
    - $O(n\ (1+poly(k)/m))$ [Sahinalp-Vishkin, FOCS'96, Cole-Hariharan, SODA'00]
  - Approximate:
    - $O(n/\varepsilon^2\ \log\ |\Sigma|\ \log\ m)$ [Karloff, IPL'93]
    - $O(n/\varepsilon^2\ \log\ m)$ [Indyk, FOCS'98]

Karloff's Algorithm

- Embed Hamming over Σ into Hamming over $\{0,1\}$:
  - Take $f: \Sigma \rightarrow \{0,1\}^{t=O(\log |\Sigma|/\varepsilon^2)}$ such that for any a,b in Σ,

$$H(f(a),f(b)) = t/2\ (1 \pm \varepsilon)$$

  - Replace each symbol a in T and P by $f(a)$, obtaining $f(T)$ and $f(P)$

a b a c b → 000 101 000 010 101

b b c → 101 101 010

L_p norms

- L_2 : Exact, in $O(n \log m)$ time
  - $||S-P||^2 = ||S||^2 + ||P||^2 - 2\ S*P$

- L_1 :
  - Exact: $O^{\sim}(n\ m^{1/2})$ [Indyk-Lewenstein-Lipsky-Porat, ICALP'04]
  - Approximate: $O(\ (m \log m + n) \log n/\varepsilon^2)$ [Indyk]

    $O(\quad n \log m \log |\Sigma|/\varepsilon^2\ )$ [Lipsky-Porat]

L_1 norm

- Imagine we have a linear mapping $A:R^m \rightarrow R^t$, $t=O(\log n/\varepsilon^2)$, such that for all P,S:

$$||P-S||_1=||AP-AS||_1\ (1\pm\varepsilon)$$

- Then we easily get an $O(n\ t \log n\ )$ algorithm:
  - Denote $A=[a_1\ a_2\ \ldots\ a_t\ ]^T$
  - Compute AP — $O(mt)$
  - For $j=1..t$, compute $a_j*T[i..i+m-1]$, $i=1\ldots n$ via FFT — $O(n\ t \log n)$
    - This gives us AS for all m-substrings S of T
  - Estimate $||P-S||_1$ for all S — $O(n\ t)$
- Faster algorithm obtained by reversing the pattern and text computation

Dimensionality reduction in L_1

- Unfortunately, such mapping A does not exist [Charikar-Sahai, FOCS'02]
- But, there are A's such that

$$||P-S||_1 = \text{median}[\,|AP-AS|\,]\,(1\pm\varepsilon)$$

with high probability [Indyk, FOCS'00]

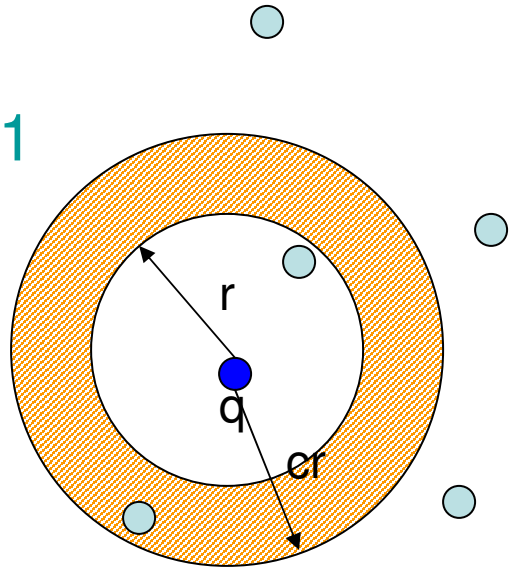
- Construction uses 1-stable distributions: $a_j * x$ has the same distribution as $z * ||x||_1$

Bonus section

- Consider the following general matching problem:
  - We have arbitrary metric (D,Σ)
  - The distance $D(P,S)=\Sigma_i D(P[i],S[i])$
- Theorem [Bourgain'85]: Any metric (D,Σ) can be embedded into $R^{O(\log |\Sigma|)}$ under L_1 with distortion $O(\log |\Sigma|)$, in time $O^{\sim}(|\Sigma|^2)$.
- Corollary: a $O(\log |\Sigma|)$-approximate algorithm for the g.m.p. [Lipsky-Porat]

Approximate Near Neighbor

- c-Approximate Near Neighbor:
  - Given: set S of N points S_i, r>0,c>1
  - Goal: build data structure which, for any query q, if there is a point $p \in P$, $||q-p||_2 \leq r$, it returns $p' \in P$, $||q-p'||_2 \leq cr$
- Can be used to solve exact NN
  - E.g., report all c-approximate NNs
  - Query time depends on the data set



Approximate NN in Hamming space

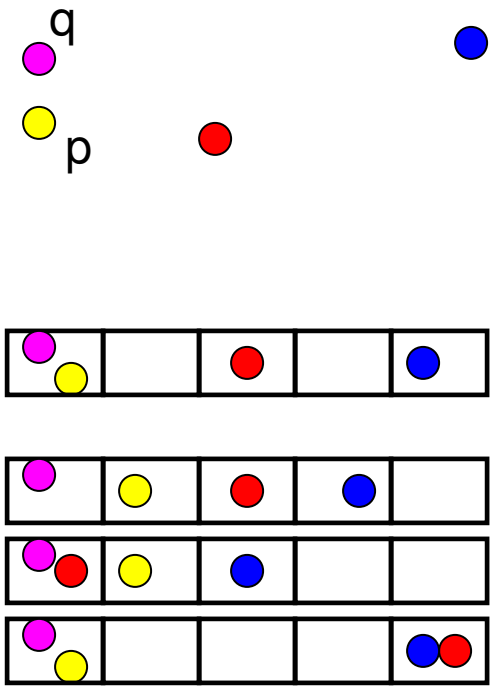
- Exact algorithms:
  - 2^m space, $O(m)$ query time
  - $O(Nm)$ time
- Approximate algorithms:
  - Space/time exponential in m [Arya-Mount-et al], [Clarkson, STOC'97], [Kleinberg, STOC'97], [Har-Peled, FOCS'02]
  - Space/time polynomial in m [Kushilevitz-Ostrovsky-Rabani, STOC'98], [Indyk-Motwani, STOC'98], [Indyk, FOCS'98],…

Approach I: Dim Reduction

- Would like to:
  - Reduce the dimension m to $t=O(\log N/\varepsilon^2)$
  - Induce only $c=(1+\varepsilon)$ distortion
- Possible for:
  - L_2 norm [Johnson-Lindenstrauss'84]
    - $\Rightarrow N^{O(\log(1/\varepsilon)/\varepsilon^2)}$ space, $O(d \log N/\varepsilon^2)$ query [Indyk-Motwani'98]
  - Hamming [Kushilevitz-Ostrowsky-Rabani'98]
    - $\Rightarrow N^{O(1/\varepsilon^2)}$ space, $O(d \log N/\varepsilon^2)$ query
- Tool: random linear map

Approach II: Locality-Sensitive Hashing [Indyk-Motwani'98]

- Idea: construct hash functions $g: \{0,1\}^m \rightarrow U$ such that for any points p,q:
  - If $D(p,q) \leq r$, then $Pr[g(p)=g(q)]$ is ~~"high"~~ "not-so-small"
  - If $D(p,q) > cr$, then $Pr[g(p)=g(q)]$ is "small"
- Then we can solve the problem by hashing



LSH for Hamming

- $g_A(p)=p_{|A}$, $|A|=t$
- Works because:

$g_A($ 0 1 0 0 1 0 1 1 0 $)=0\ 0\ 1$

$g_A($ 0 1 0 0 1 0 0 1 0 $)=0\ 0\ 1$

$g_A($ 0 0 0 1 0 0 0 1 0 $)=0\ 0\ 0$

- However, t is large, so

  $p \rightarrow p_{|A} * (a_1,...,a_t) \bmod M$

0 1 0 0 1 0 1 1 0 *
$a_1$0 $a_2$0$a_3$0 0 0 0

- Can show #hash tables $= N^{1/c}$
- $O(N^{1+1/c})$ space, $O(mN^{1/c} \log N)$ query time

All m-substrings version

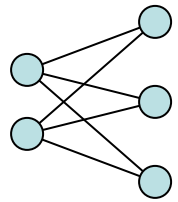
- Can
  - Generate N-m+1 substrings of T[1…N]
  - Use LSH algorithm
- Drawback: $O(m N^{1+1/c})$ preprocessing time
- But, we hash all substrings of T using FFT
  - $O(N \log m)$ time per hash function
  - $O(N^{1+1/c} \log m)$ time total
- Other optimizations possible [Buhler, RECOMB'02,…]

Edit distance

- Many algorithms for the exact problem
- Approximation algorithms ?
- Embeddings ?

Embeddings of Edit Distance

- ED cannot be embedded into L_1 with distortion $\leq 3/2$
  [Andoni-Deza-Gupta-Indyk-Raskhodnikova, SODA'02]



- ED over strings of length $\leq m$ can be embedded* into L_1 with distortion $O(m^{\rho})$ [Bar-Yossef-Jayram-Krauthgamer-Kumar, FOCS'04]

Block Edit Distance

- If we allow block operations (each with unit cost):
  - Move: ababcd $\rightarrow$ cdabab
  - Copy: abcd $\rightarrow$ abcdab (plus the inverse op)
  - Etc.
- Then BED can be embedded into L_1 with distortion $O(\log m \log^* m)$

  [Cormode-Paterson-Sahinalp-Vishkin, SODA'00, Muthukrishnan-Sahinalp, STOC'00, Cormode-Muthukrishnan, SODA'02]

Implications

- BED:
  - $O(\log m \log^* m)$-approximate NN with $O(N^{1.1})$ space, poly(m) query [Muthukrishnan-Sahinalp'00]
  - $O(\log m \log^* m)$-approximate pattern matching in $O^{\sim}(n+m)$ time [Cormode-Muthukrishnan'02]
- ED:
  - $O(m^{\rho})$ -approximate NN with $O(N^{1.1})$ space, poly(m) query for some $\rho>0$ [Bar-Yossef et al'04]

    Known: $O(m^{\rho})$-approximate NN with $O(N^{2^{1/\rho}})$ space for any $\rho>0$ [Indyk, SODA'04]
  - $O(m^{\rho})$-approximate pattern matching in $O^{\sim}(n+m)$ time

Edit and Hamming Distances

- Want to find patterns modified by:
  - k insertions/deletions (indels)
  - I substitutions
  - k << I
- Can find a substring [Badoiu-Indyk, SODA'04]:
  - With k indels, $(1+\varepsilon)$I substitutions,
  - In time $O(n \text{ poly}(1/\varepsilon + k + \log n))$
- Method: Extend the O(nk)-time algorithm:
  - Instead of finding longest T[i…j] matching prefix of P, find the longest T[i…j] matching prefix of P approximately
  - Use poly(log m+1/ε) data structure from [Indyk-Koudas-Muthukrishnan, VLDB'00]

Conclusions

- Examples of embeddings:
  - General metrics into L_1
  - Concrete metrics into L_1
  - Dimensionality reduction
- Applications to problems:
  - Pattern matching
  - Near Neighbor

Open Problems

- Near neighbor:
  - Improve the $O(m\, n^{1/c})$ query time (but keep small space)
    - Recent (small) improvement for L_2 norm [Datar-Immorlica-Indyk-Mirrokni, SoCG'04]
  - Better space bound for data set induced by substrings of T of arbitrary length m
    - Preprocessing for all m's gives $O(n^{1+1+1/c})$ space
- General pattern matching tradeoff:
  - Exact, $O(|\Sigma|\, n \log n)$ time
  - $\log |\Sigma|$-approximate, $O^{\sim}(n)$-time

Open Problems

- Better embeddings (or lower bounds) for ED or BED into L_1
- Better NN for k indels, l substitution, $k<<l$

The End – Thank You!