Low-distortion embeddings and data structures

Piotr Indyk
MIT
What this talk is about

• Low-distortion embeddings:
  – Metrics \((X,D),(X',D')\)
  – Mapping \(f: X \rightarrow X'\)
  – Want
    \[D(p,q) \leq D'(f(p),f(q)) \leq c \ D(p,q)\]

• Data structures:
  – Support some operations on a data set \(P\)
  – Simple example for \(P \subseteq \{1\ldots M\}\):
    • Insert \((p)\): inserts \(p\) into \(P\)
    • Delete \((p)\): deletes \(p\) from \(P\)
    • Distinct-Count: returns the number of distinct elements in \(P\)
Menu

• Nearest Neighbor
  – In high dimensional $l_p^d$ spaces (focus on $p=2$)
  – In other metrics (Hausdorff, EMD, edit)
• Data structures with sub-linear storage:
  – Distinct-Count and more
• Distance oracles
  – Given $p,q$, report $D(p,q)$
  – Sub-quadratic storage
  – Very fast distance computation

All algorithms are:
  – Approximate
  – Randomized (can work with probability, say, $2/3$)
Nearest neighbor

• Given: a set $P$ of $n$ points in $\mathbb{R}^d$
• **Nearest Neighbor**: for any query $q$, returns a point $p \in P$ minimizing $||p-q||$
• **$r$-Near Neighbor**: for any query $q$, returns a point $p \in P$ s.t. $||p-q|| \leq r$ (if it exists)
Nearest Neighbor: Motivation

- Learning: nearest neighbor rule
- Database retrieval
- Vector quantization, a.k.a. compression
The case of $d=2$

- Compute Voronoi diagram
- Given $q$, perform point location
- Performance:
  - Space: $O(n)$
  - Query time: $O(\log n)$
The case of $d>2$

- Voronoi diagram has size $n^{O(d)}$
- We can also perform a linear scan: $O(dn)$ time
- That is pretty much all what is known (for the exact problem)
Approximate Near Neighbor (NN)

- **c-Approximate r-Near Neighbor**: build data structure which, for any query q:
  - If there is a point \( p \in P \), \( ||p-q|| \leq r \)
  - It returns \( p' \in P \), \( ||p-q|| \leq cr \)

- **Reductions**:
  - c-Approx Nearest Neighbor reduces to c-Approx Near Neighbor
    - Query time: multiplied by \( \log n \)
    - Space: multiplied by \( \log^{O(1)} n \)

[Indyk-Motwani’98; Kushilevitz-Ostrovski-Rabani’98; Har-Peled’01]
Johnson-Lindenstrauss

- **JL**: Any n-point subset $X$ of $l_2^d$ embeds into $l_2^{d'}$ with distortion $1+\varepsilon$ for $d'=O(\log n/\varepsilon^2)$

- **JL’**: There is a distribution over mappings $A: l_2^d \rightarrow l_2^{d'}$ such that, for any $x \in l_2^d$:

  $$\Pr[||x|| \leq ||Ax|| \leq (1+\varepsilon)||x||] \geq 1 - \exp(\varepsilon^2 d')$$

- Clearly, JL’ $\Rightarrow$ JL. But all proofs of JL imply JL’ as well.
- All applications mentioned in this talk require JL’, since some/all vectors $x$ are not known in advance.
(1+\(\varepsilon\))-approximate r-NN with space polynomial in \(n\)

1. Map \(A: \mathbb{R}^d \to \mathbb{R}^{d'}, d' = O(\log n/\varepsilon^2)\)

2. Construct r-NN data structure:
   - Space: \(n(1/\varepsilon)^{O(d')}\)
   - Query: \(O(d')\)

3. To find approx r-NN of \(q\), query \(Aq\)

Overall:
- Space: \(n^{O(\log(1/\varepsilon)/\varepsilon^2)}\) (better exponent of \(O(1/\varepsilon^2)\) [KOR’98])
- Query: \(O(d \log n/\varepsilon^2)\) (improved via FJLT – [Ailon-Chazelle’06])
Metrics

• Distances between multi-sets of points in $\mathbb{R}^t$
  – Hausdorff metric:
    \[ DH(A,B) = \max_{a \in A} \min_{b \in B} ||a-b|| \]
    \[ H(A,B) = \max[DH(A,B), DH(B,A)] \]
  – Earth Mover Distance
    \[ EMD(A,B) = \min_{\pi: A \rightarrow B} \sum_{a \in A} ||a-\pi(a)|| \]

• Distances between strings of symbols:
  – ED(s,s′): min #ins/del of symbols
  – BED(s,s′): block operations as well
    (block move, block copy and reverse operations)
    \[ ED(\text{abracadabra}, \text{dabra}) = 6 \]
    \[ BED(\text{abracadabra}, \text{dabra}) = 3 \]

• Can obtain algorithms for such metrics by
  embedding them into normed spaces
# Embeddings

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Dist.</th>
<th>Dim.</th>
<th>Paper</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hausdorff over m-subsets of ${1..D}^t$</td>
<td>$l_\infty$</td>
<td>$1+\varepsilon$</td>
<td>$m^2(1/\varepsilon)^t \log^2 D$</td>
<td>Farach-Colton-Indyk’99</td>
</tr>
<tr>
<td>EMD over ${1..D}^t$</td>
<td>$l_1$</td>
<td>$\log D$</td>
<td>$D^{O(1)}$</td>
<td>Charikar’02; Indyk-Thaper’03</td>
</tr>
<tr>
<td></td>
<td>$l_1$</td>
<td>$&gt;(\log D)^{1/2}$</td>
<td></td>
<td>Naor-Schechtman’06</td>
</tr>
<tr>
<td></td>
<td>$l_1$</td>
<td>$&gt;t$</td>
<td></td>
<td>Khot-Naor’05</td>
</tr>
<tr>
<td>Block edit distance over d-length strings</td>
<td>$l_1$</td>
<td>$\approx \log d$</td>
<td></td>
<td>Muthu-Sahinalp’00; Cormode-Muthu’02</td>
</tr>
<tr>
<td>Edit distance over d-length strings</td>
<td>$l_1$</td>
<td>$\exp[ (\log d)^{1/2} ]$</td>
<td></td>
<td>Ostrovski-Rabani’05</td>
</tr>
<tr>
<td></td>
<td>$l_1$</td>
<td>$&gt;\log d$</td>
<td></td>
<td>Khot-Naor’05; Krauthgamer-Rabani’06</td>
</tr>
</tbody>
</table>
Sub-linear storage
Norm estimation

- Norm estimation:
  - Initially: $x=0$
  - Stream elements: $(i,b)$, $i=1\ldots d$, $b \in \{-d^{O(1)} \ldots d^{O(1)}\}$
  - Interpretation: $x_i = x_i + b$
  - Want to maintain $||x||_p$
  - ...using little space, i.e., only $\log^{O(1)} d$ bits

- Why? Examples:
  - $||x||_p = \sum_i x_i^p = \text{#non-zero coordinates in } x$, as $p \to 0$
  - Maintains the number of distinct elements under
    - Insertions: $(i,1)$
    - Deletions: $(i,-1)$
Dimensionality reduction

- Store $Ax$ instead of $x$
- Key observation: can update $Ax$ under updates to $x$
- Recover $(1 \pm \epsilon)\|x\|_p$ from $Ax$ (with prob. $1-1/d$)
- Issue: cannot store $A$, must be “pseudorandom”
- Algorithms:
  - $p=2$: [Alon-Matias-Szegedy’96]
    - Estimator: median\[(A_1x)^2 + \ldots + (A_c x)^2, (A_{c+1}x)^2 + \ldots + (A_{2c} x)^2, \ldots]\] $^{1/2}$
    - $c=1/\epsilon^2$, $k=c \log d$
    - $A$: constructed from 4-wise independent random variables
  - $0 < p \leq 2$: [Indyk’00]
    - Estimator: median\[(A_1x), \ldots, (A_k x)\]
    - $A$: constructed using Nisan’s PRG
What else?

• Maintaining geometric statistics (MST cost, min matching cost) of sets of points
  – E.g., we can maintain $\text{EMD}(A,B)$ under changes to $A,B$
    • $\text{EMD}(A,B)$ into $l_1$ with dist. $\log D$
    • Can maintain $l_1$ norm
    • Compose

• Maintaining a sparse approximation of a vector $x$
Sparse Approximations

• View $x$ as a function $x: \{1 \ldots d\} \rightarrow \{-d^{O(1)} \ldots d^{O(1)}\}$

• Approximate it using simpler functions
  – Linear combinations of at most $B$ vectors in some given basis (Fourier, wavelets, etc)
  – Piecewise constant function $h$, with $B$ pieces (buckets)
  – Etc..

• Goal: find $h$ s.t. $\|x-h\|_2 \leq (1+\varepsilon)\|x-h_{OPT}\|_2$
Results

• [Gilbert-Guha-Indyk-Kotidis-Muthukrishnan-Strauss’02] :
  – Under increments/decrements of $x$ maintains piecewise constant $h$ with $B$ pieces such that
    $$\|x-h\|_2 \leq (1+\varepsilon)\|x-h_{OPT}\|_2$$
  – Space: $\text{poly}(B,1/\varepsilon,\log n)$
  – Time: $\text{poly}(B,1/\varepsilon,\log n)$
General Approach

• Maintain sketches $Ax$ of $x$
• This allows us to estimate the error of any approximation $h$, via $||x-h|| \approx ||Ax-Ah||$
• Construct $h$ (“invert” the sketch):
  – Enumeration – exponential in $B$
  – Greedy
  – Dynamic Programming
Compressed sensing

- [Donoho’05; Candes-Romberg-Tao’06; Rudelson-Vershynin’05; …….]
  - Consider $x$ which are $B$-sparse (with respect to any fixed basis) or some generalizations involving noise
  - Show that there are mappings $A: \mathbb{R}^d \rightarrow \mathbb{R}^k$ such that, for any $x$, given $Ax$, one can reconstruct $x$
    - Gaussian matrix: $k=O(B \log(d/B))$
    - Fourier matrix: $k=O(B \log^0(1) d)$
  - Properties can be proved using JL lemma [Baraniuk-Davenport-DeVore-Wakin’06]
  - Reconstruction: minimize $\|z\|_1$ s.t. $Az=Ax$
    - Can be done using linear programming

- See [http://www.dsp.ece.rice.edu/cs/](http://www.dsp.ece.rice.edu/cs/) for more info
Distance oracles
Metric compression

- Compressed representation of a metric $M = (X, D)$, $|X| = n$:
  - Spanners [Peleg, etc]: sparse graph $G = (X, E)$ such that $M$ \(c\)-embeds into a metric induced by $G$
  - Can guarantee $|E|/n \leq n^{\beta(c)}$, for $\beta(c) = 1/\lfloor(c+1)/2\rfloor \approx 2/c$
- Fast distance computation [Cohen’94]:
  - Approximate $D(p, q)$ in time $n^{\beta(c)}$
- Can get the same result from metric embeddings into $l_\infty$ [Matousek’96]
- [Thorup-Zwick’01]: “Distance oracles”
  - Approximate $D(p, q)$ in time $O(c)$
- [Mendel-Naor’06]: “Ramsey partitions”
  - Approximate $D(p, q)$ in time $O(1)$