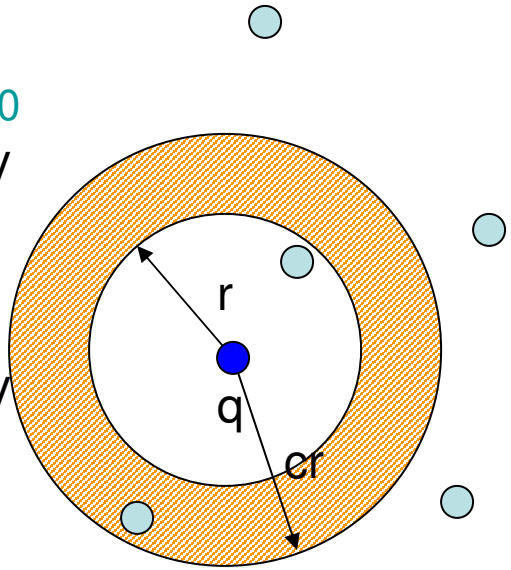
Efficient Similarity Search in High Dimensions

Piotr Indyk
MIT

(Approximate) Near Neighbor

- Near neighbor:
  - Given:
    - A set P of points in a metric space $M=(X,D)$, $r>0$
  - Goal: build data structure which, for any query q, returns a point $p \in P$, $D(p,q) \le r$ (if it exists)



- c-Approximate Near Neighbor:
  - Goal: build data structure which, for any query q:
    - If there is a point $p \in P$, $D(p,q) \le r$
    - it returns $p' \in P$, $D(p,q) \le cr$

- Can be used to solve exact near neighbor
  - Set $c=1$, or
  - Report all c-approximate NNs (query time depends on the data set)
- Can be used to solve (approximate) nearest neighbor

Talk overview

- NN in R^d spaces: Locality-Sensitive Hashing
- NN in more general metrics: Earth-Mover Distance
- NN in general metrics with "low intrinsic dimension"

NN in R^d

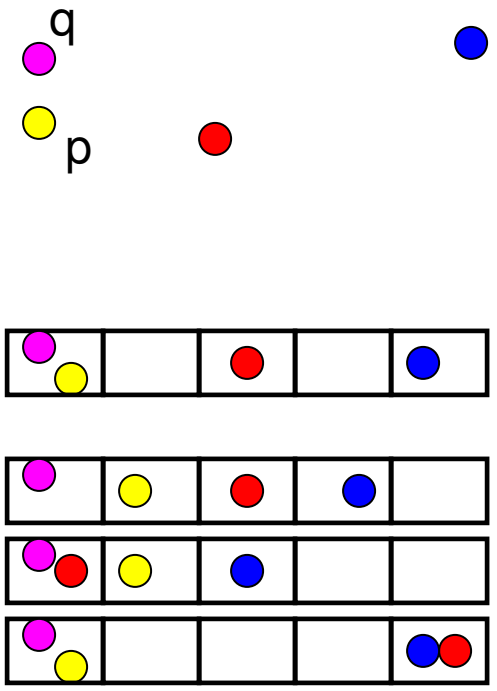
NN in R^d : history

- Exact algorithms use
  - Either $n^{O(d)}$ space,
  - Or $O(dn)$ time
- Approximate algorithms:
  - Space/time exponential in d [Arya-Mount-et al], [Kleinberg'97], [Har-Peled'02]
  - Space/time polynomial in d [Kushilevitz-Ostrovsky-Rabani'98], [Indyk-Motwani'98], [Indyk'98],…

Locality-Sensitive Hashing

[Indyk-Motwani'98]

- Idea: construct hash functions $g: R^d \rightarrow U$ 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

- A family H of functions $h: I_s^d \rightarrow U$ is called (P_1,P_2,r,cr)-sensitive, if for any p,q:
  - if $||p-q||_s < r$ then $Pr[\ h(p)=h(q)\ ] > P_1$
  - if $||p-q||_s > cr$ then $Pr[\ h(p)=h(q)\ ] < P_2$
- We hash using $g(p)=h_1(p).h_2(p)\ldots h_k(p)$
  - Intuition: amplify the probability gap
- We can solve a c-approximate NN with:
  - Number of hash functions: n^ρ , $\rho = \log_{1/P2}(1/P_1)$
    - Query time: $O(d\ n^\rho \log n)$
    - Space: $O(n^{\rho+1} + dn)$

LSH for Hamming metric [IM'98]

- Functions: $h(p)=p_i$, i.e., the i-th bit of p
- We have

$$\Pr[\, h(p)=h(q)\,] = 1-D(p,q)/d$$

- This gives exponent $\rho=1/c$
- Used for genetic sequence retrieval, motif finding, video sequence retrieval, compression, web clustering, pose estimation, etc.

Other norms

- Can embed l_1^d with coordinates in $\{1 \ldots M\}$ into dM-dimensional Hamming space
- Can embed l_2^d into $l_1^{O(d)}$
- In practice, not so easy

New LSH scheme

- New LSH scheme [Datar-Immorlica-Indyk-Mirrokni'04]
- Works directly in l_s spaces (unlike [IM'98])
- Bounds on ρ (recall query time is $O(dn^{\rho})$)
  - $\rho < 1/c$ for l_2 (improves on [IM'98])
  - $\rho \approx 1/c$ for l_1
  - $\rho \approx 1/c^s$ for l_s, $s<1$
- Experiments indicate that the new LSH is quite fast
  - Order of magnitude speedup over (approximate) kd-tree

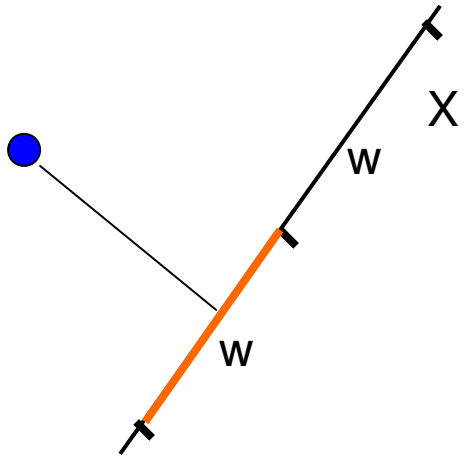
Plan

- Describe the new scheme
- Sketch the analysis
- Experiments

LSH scheme

Define $h_{X,b}(p)=\lfloor (p*X+b)/w \rfloor$:

- $w \approx r$
- $X=(X_1 \ldots X_d)$, where X_i is chosen from "s-stable" distribution
- I.e., $p*X$ has same distribution as $||p||_s$ Z, where Z is s-stable
- For l_2, Gaussian distribution is stable



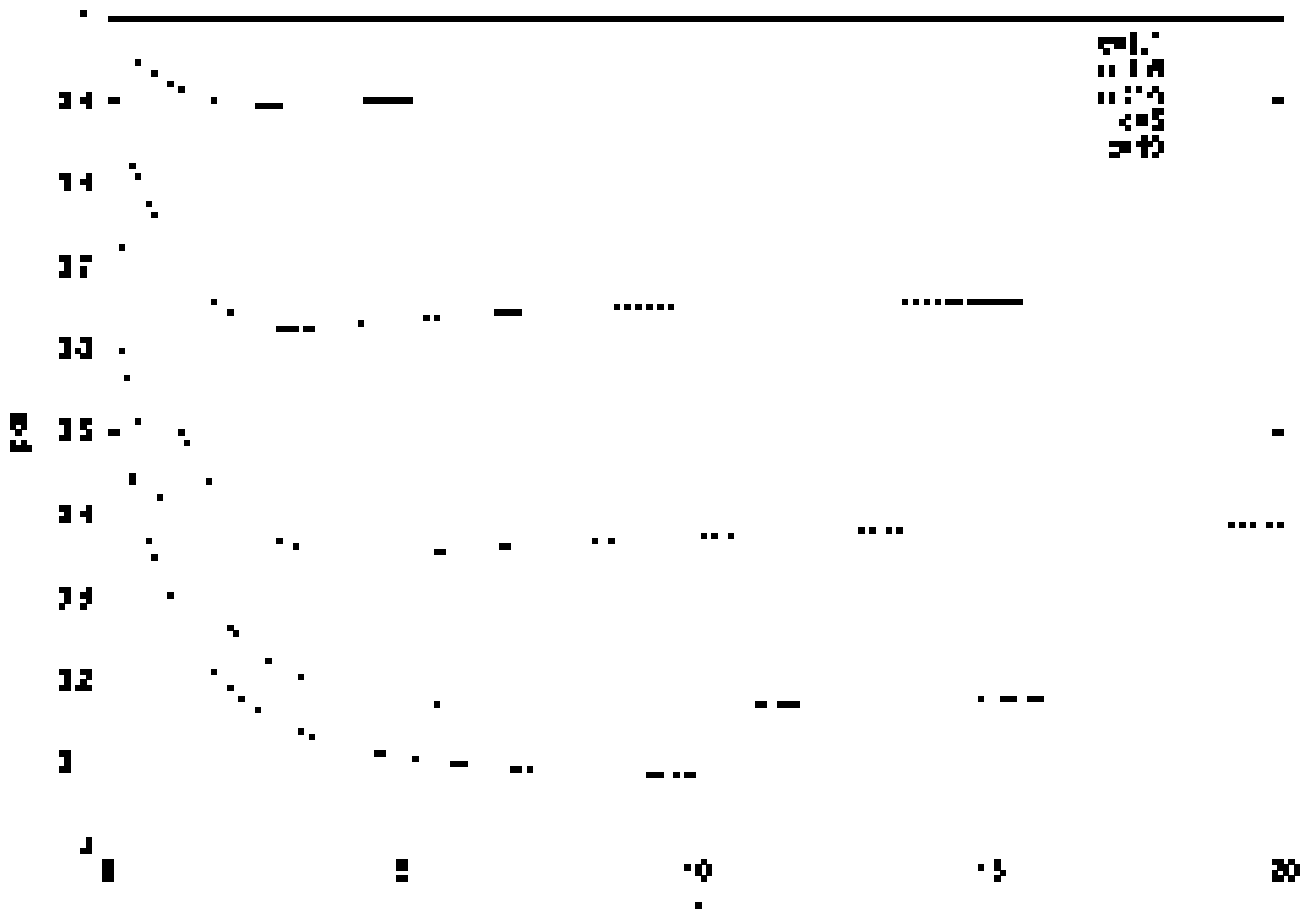
Analysis

- Need to:
  - Compute $\Pr[h(p)=h(q)]$ as a function of $||p-q||$ and w; this defines P_1 and P_2
  - For each c choose w that minimizes

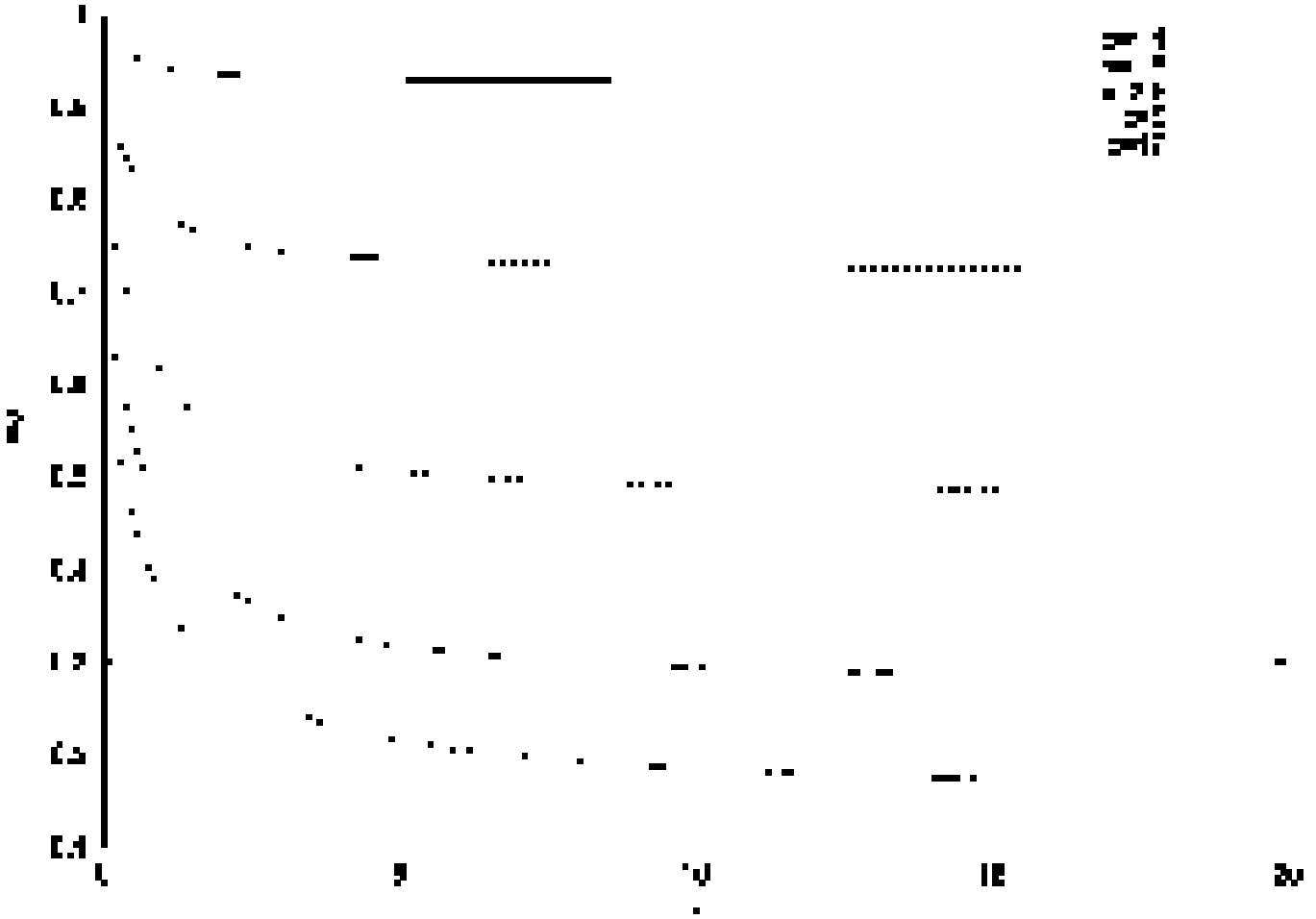
$$\rho=\log_{1/P2}(1/P_1)$$

- Method:
  - For l_2: computational
  - For general l_s: analytic

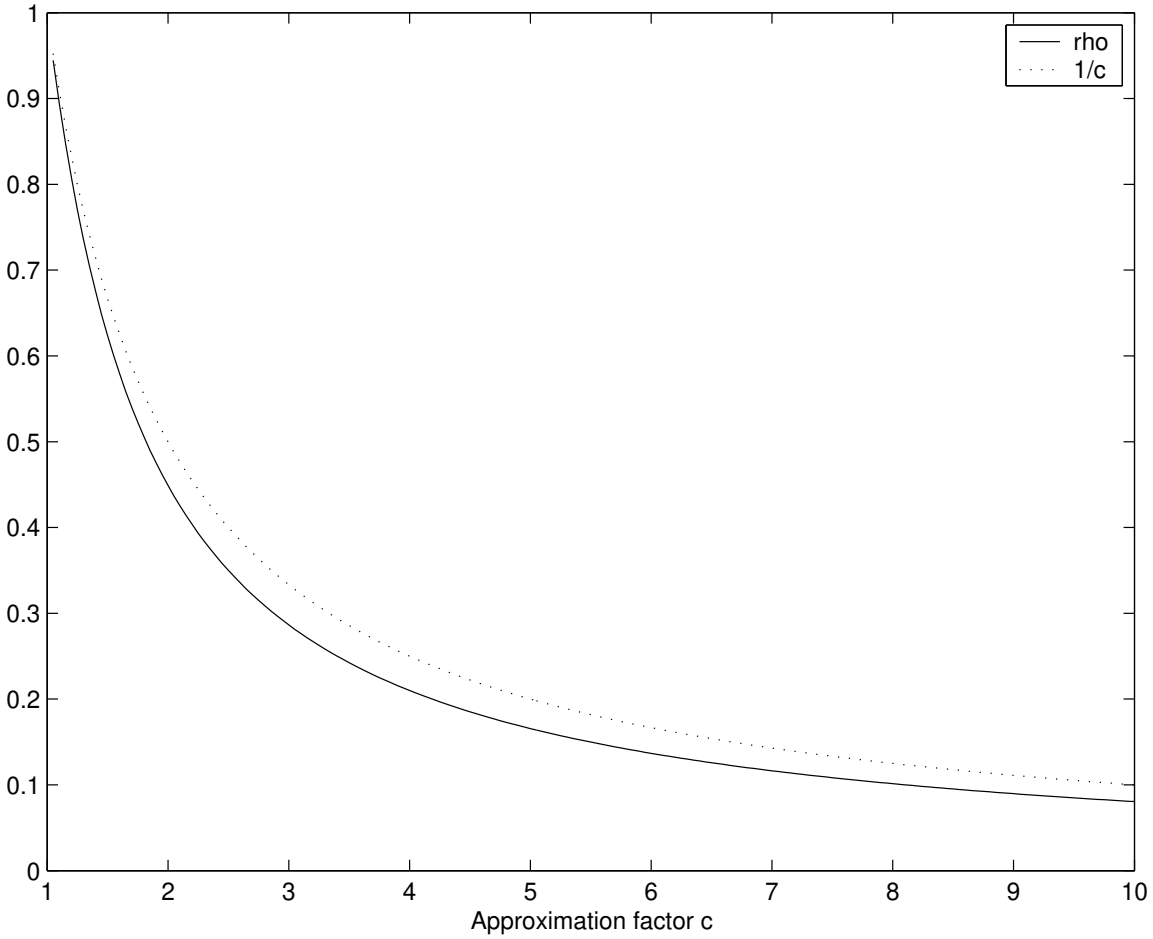
ρ(w) for various c's in l_2



Compare to l_1



ρ(c) for l_2


rho
1/c
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1
2
3
4
5
6
7
8
9
10
Approximation factor c

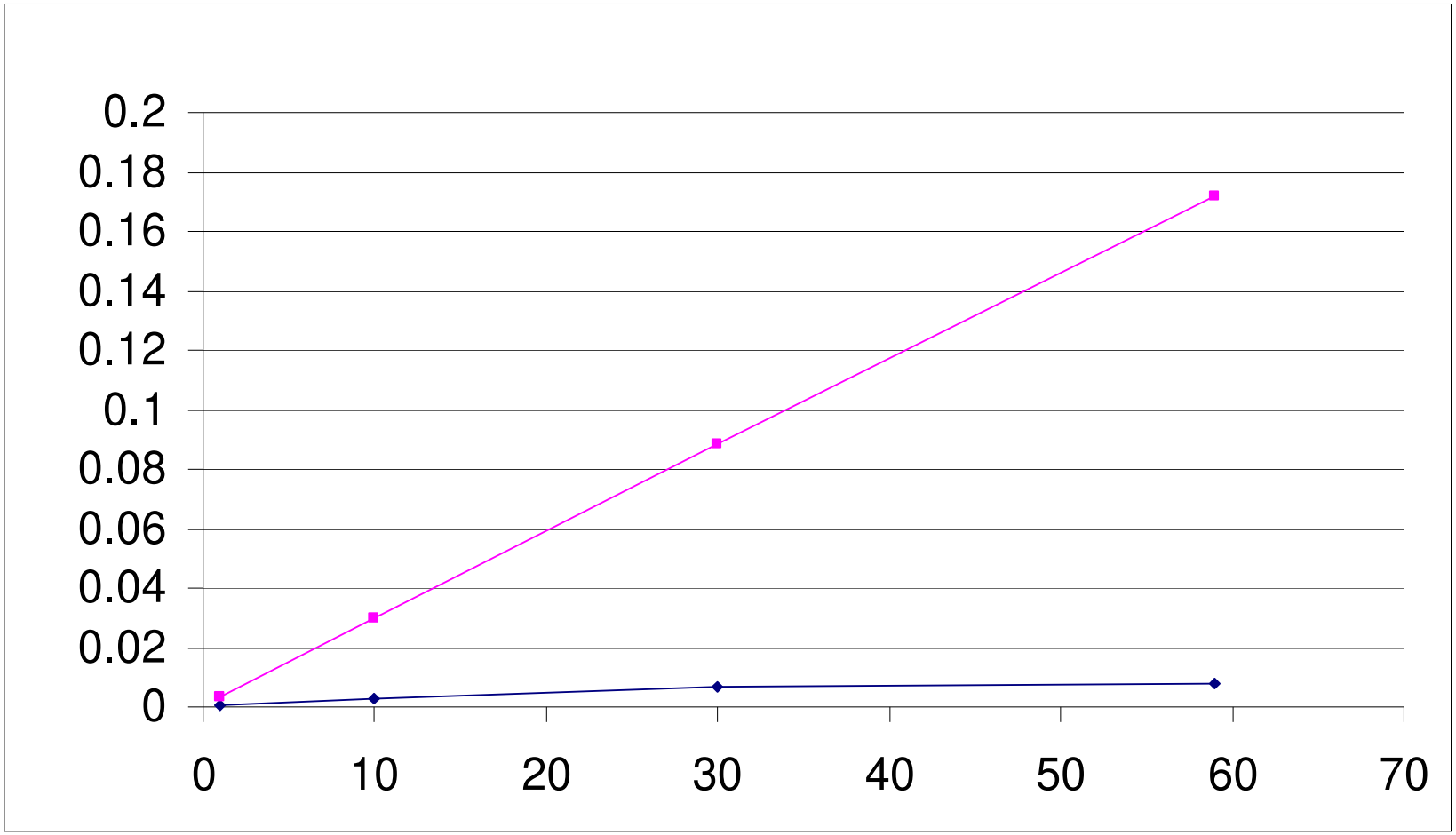
Experiments

- Exact near neighbor:
  - User sets r and P
  - Program finds parameters k,l,w so that:
    - Probability of NOT reporting a point within distance r is <P
    - Expected query time is minimized
- Nearest neighbor: set r to accommodate "almost all" queries

Data sets

- In proceedings: planted near neighbor
- This talk: NIST OCR data
  - d=784
  - n=60,000
- LSH:
  - Error prob=0.1
  - Radius r set to 0.55
- Kd-tree (ANN package, 1+ε approx.):
  - ε=0…100%

LSH vs kd-tree



Caveats

- For ANN, setting ε=1000% results in:
  - Query time comparable to LSH
  - Correct NN in about 65% cases, small error otherwise
- However, no guarantees
- LSH eats much more space:
  - LSH: 1.2 GB
  - Kd-tree: 360 MB

LSH

- Send me an e-mail for the code

NN Under Un-normed Metrics

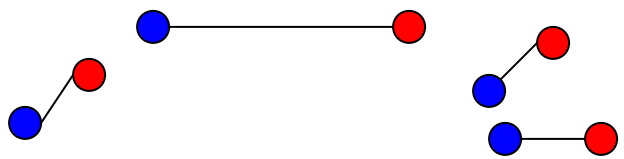
Example I: Hausdorff Metric

- Given: two sets $P,Q \subseteq R^d$
- $DH(P,Q)=\max_{p\in P} \min_{q\in Q} ||p-q||$
- $H(P,Q)=\max[\ DH(P,Q)\ ,\ DH(Q,P)\ ]$

- Nearest Neighbor: build a data structure for $S=\{P_1 \ldots P_n\}$ that given Q finds NN of Q in S

Example II: Earth-Mover Distance

- Given: two (multi)sets $P,Q \subset R^d$, $|P|=|Q|$
- EMD(P,Q)=min weight matching between P and Q



NN in Un-Normed Metrics

- Embeddings: Construct a mapping f:
  - f maps each "shape" set P into a vector $f(P)$
  - For any P,Q, we have

$$D(P,Q) \leq ||f(P)-f(Q)|| \leq c\, D(P,Q)$$

  - Search for NN of $f(Q)$ in $f(P_1)\ldots f(P_n)$
  - This gives c-approximate NN of Q in $P_1\ldots P_n$

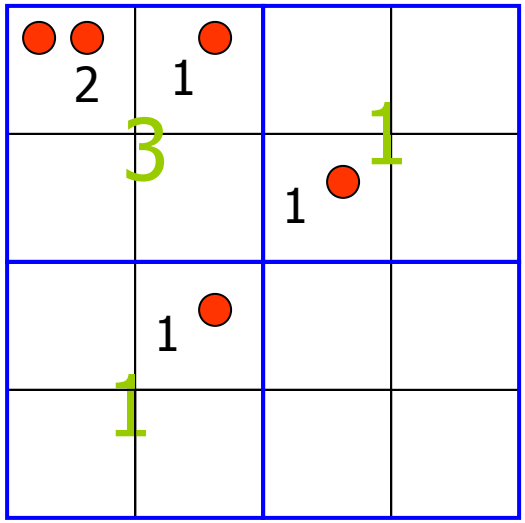
Approximate NN via Embeddings

- Known results:
  - [Bourgain]: Any metric into l_2 with distortion logarithmic in the metric size
    - Problem: needs whole metric in advance
  - [FCI'99]: Hausdorff over low-dimensional sets into low dimensional l_∞
  - [CPSV'00,MS'00,CM'02]: Block-edit distance into l_1
- This talk: EMD into l_1^d

Part I: EMD into l_1

- Assume $P \subseteq \{1,\ldots,\Delta\}^2$
- Impose square grids $G_{-1}\ldots G_k$, with side lengths $2^{-1},2^0, \ldots, 2^k = \Delta$, shifted at random.
- For each square cell c in G_i, let $n^i_P(c)$ be the number of points in $|c \cap P|$.
- Embedding: P is mapped to

$$f(P) = 2^{-1}n^{-1}_P, 2^0n^0_P \ldots 2^kn^k_P$$

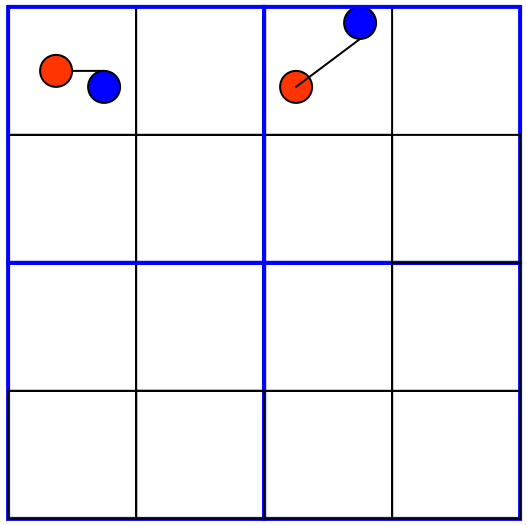


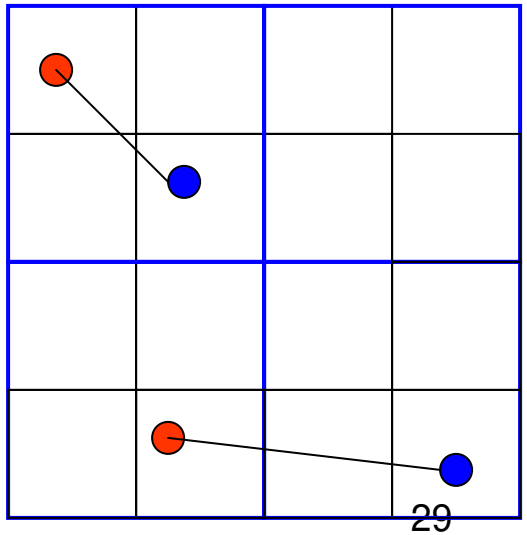
Guarantees

- Theorem:
  - EMD(P,Q) < O($||f(P)-f(Q)||_1$)
  - E[$||f(P)-f(Q)||_1$] = O(log Δ) EMD(P,Q)
- Due to:
  - Charikar'02+Kleinberg,Tardos'99+Bartal'96+ Peleg+Goel [personal communication]
  - Indyk,Thaper'02, Varadarajan'02

Proof intuition

- EMD(P,Q) small:
  - Most points in P are close to the corresponding points in Q
  - Corresponding points fall to the same cell
  - Counts cancel out: $||f(P)-f(Q)||_1$ small
- EMD(P,Q) large:
  - Many points in P are far from the points in Q
  - Corresponding points fall to different cells
  - Counts do not cancel out
  - $||f(P)-f(Q)||_1$ large

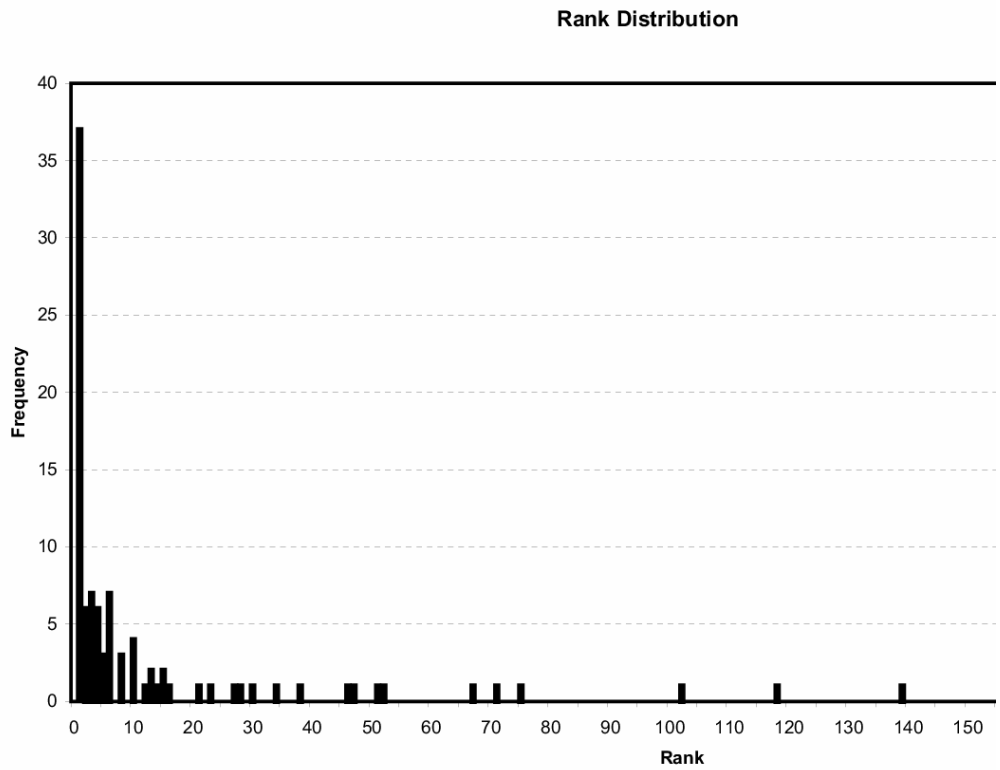




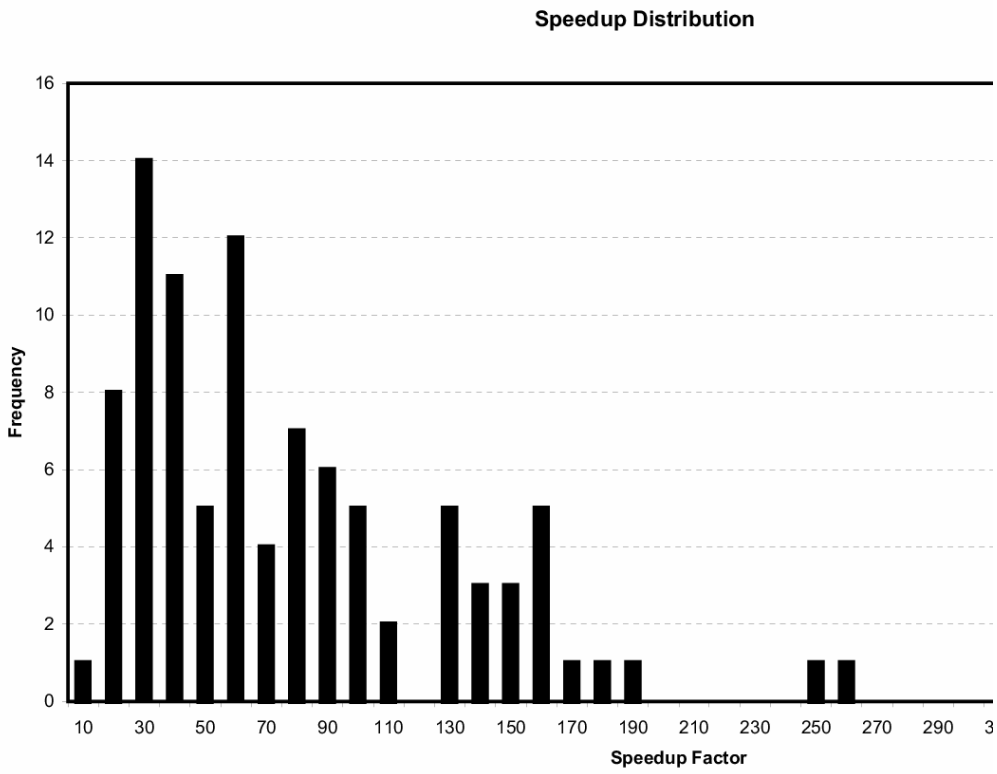
How does this work in practice?

- Data: color histograms of 20,000 Corel-Draw images:
  - Each pixel in an image is a point in 3D color space
  - Image represented by a (multi)set of pixels
- 100 queries
- Parameters:
  - Probability of failure set to 10%
  - Embedding done 5 times per query
  - Parameters c and r set by hand
- Compare our approximate NN to the exact NN (w.r.t. EMD)

NN quality: rank



Speedup


Speedup Distribution
Frequency
16
14
12
10
8
6
4
2
0
10
30
50
70
90
110
130
150
170
190
210
230
250
270
290
Speedup Factor

More info

- Results appear in [Indyk-Thaper, ICCV'03 workshop]
- See [Darrell-Grauman, CVPR'04] for more applications

Metrics With Low Intrinsic Dimension

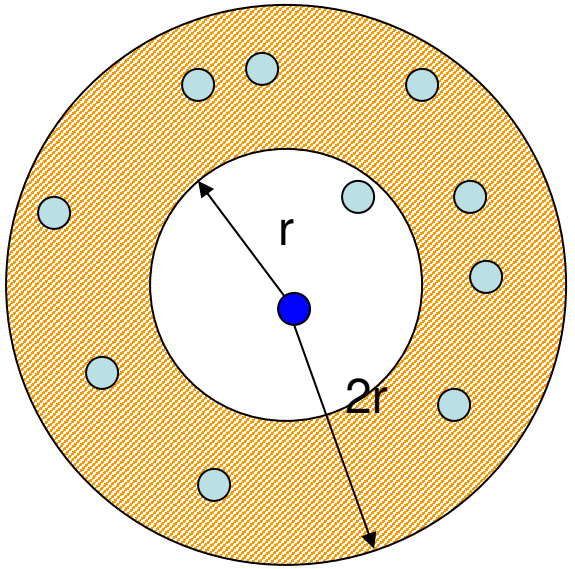
Intrinsic Dimension

- General metrics are hard
- Often the “real” dimension of the data is low:
  - Data lives on a low-dimensional manifold
- Questions:
  - How to quantify the intrinsic dimensionality ?
  - How to exploit it ?

Intrinsic Dimension I

- Bounded growth:

  LOG($\max_{p,r}$ B(p,2r)/B(p,r))

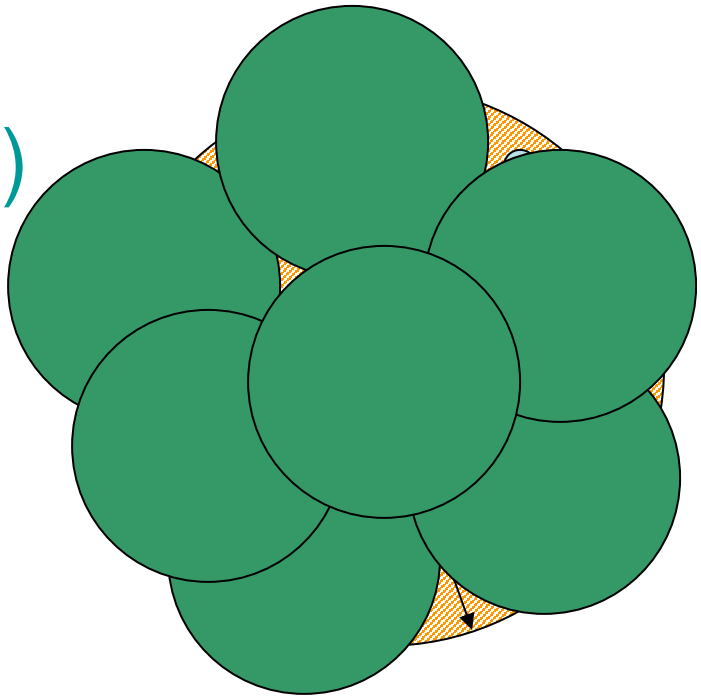


Intrinsic Dimension II

- Doubling dimension:

LOG($\text{Max}_{p,r}$ Cover(r,B(p,2r)))

  - Always not (much) greater than growth dimension
  - …Opposite does not hold
  - Subsets of R^d have O(d) doubling dimension



Results: General Metrics

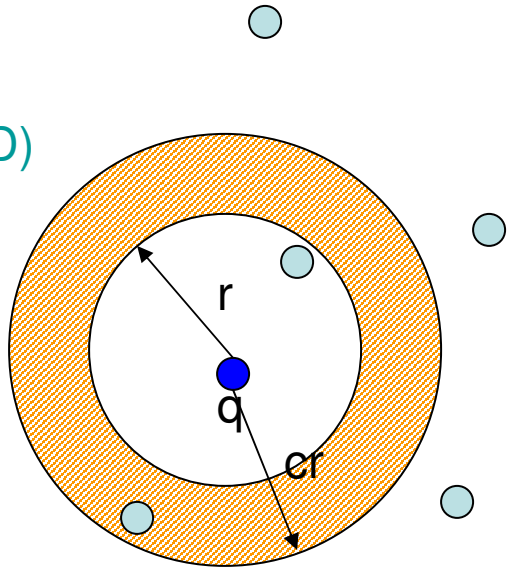
- Nearest Neighbor Algorithms with:
  - Polynomial (in n) space
  - Query time: $\log^{O(1)} n * \exp(O(\dim))$
  - Guarantees:
    - Bounded growth: exact answer
    - Doubling: approximate answer
- See [Clarkson'97, Karger-Ruhl'02, Krauthgamer-Lee'04, Beygelzimer-Kakade-Langford'04]
- Code: web pages of Ken Clarkson and John Langford

Results: Euclidean Norm

- For bounded growth metrics, LSH find exact near neighbor using exp(O(dim)) hash functions [DIIM'04]
- One can reduce the dimension to O(dim) so that an (approximate) nearest neighbor of each point is preserved with constant probability [Indyk'04]

Recall:Approximate Near Neighbor

- c-Approximate Near Neighbor:
  - Given:
    - A set P of points in a metric space M=(X,D)
    - r>0,c>1
  - Goal: build data structure which, for any query q:
    - if there is a point $p \in P$, $D(p,q) \leq r$
    - it returns $p' \in P$, $D(p,q) \leq cr$



- Can be used to solve exact near neighbor
  - E.g., report all c-approximate NNs
  - Query time depends on the data set
- Can be used to solve approximate nearest neighbor

Conclusions

- Algorithms for Euclidean Norm
  - LSH
- Algorithms for Specific Metrics
  - Embeddings
- Algorithms for General Metrics
  - Growth dimension
  - Doubling dimension
- Codes available