On the Difficulty of Nearest Neighbor Search

Junfeng He*, Sanjiv Kumar+ and Shih-Fu Chang*, Columbia University* and Google Research+

Introduction

Large Scale NN Search in the Era of Big Data

- Big data: web multimedia, enterprise data centers, mobile/surveillance sensor systems, network nodes, etc.…
- Large scale NN search for many big data applications
  - Retrieval from massive data such as multimedia search
  - Build neighborhood graphs for learning tasks like spectral clustering
  - ….

Large Scale NN Search Methods

- Exhaustive NN search: prohibitively expensive for large scale data
- Recently many approximate NN search Methods
  - Tree based methods: kd-tree, metric tree,…
  - Hashing based methods: Locality Sensitive Hashing (LSH), spectral hashing, …

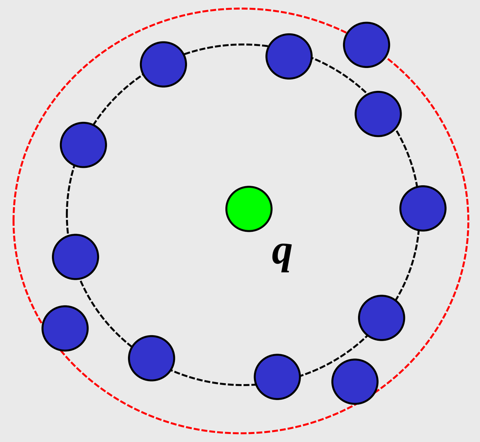
A More Fundamental Problem

- **How to measure the difficulty of a given data set for NN search, independent of NN search Methods?**
- **Moreover, what data properties affect the difficulty, and how?**

Difficulty Measure– Relative Contrast

A Toy Example

If we can not differentiate NN point from other points, NN search is not meaningful!

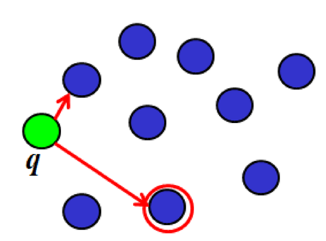


A Concrete Measure -- Relative Contrast

Relative Contrast

$$C_r = \frac{D_{random}}{D_{nn}} = \frac{E_x[D(q,x)]}{D(q,x_{nn})}$$

$$\boxed{C_r = \frac{E_{q,x}[D(q,x)]}{E_q[D(q,x_{nn})]}}$$



High Relative Contrast → more meaningful search

If $C_r \to 1$, search not meaningful

Normalized Variance σ'

– Given a database $X = \{x_i\}_{i=1}^n\ x \in \Re^d$, a query q, and a distance metric (say L_1),

$$D(q,x) = \sum_{j=1}^{d} |q^j - x^j| \Longrightarrow D = \sum_{j=1}^{d} D_j$$

– Let dimensions be *i.i.d.* with $E[D_j] = \mu_d$, $var[D_j] = \sigma_d^2$

From central limit theorem for large enough d

$$D \sim N(\mu, \sigma^2) \qquad \mu = d\mu_d \quad \sigma^2 = d\sigma_d^2$$

– If data is scaled such that $\mu' = 1$, then new variance

$$\sigma'^2 = \frac{\sigma^2}{\mu^2} = \frac{1}{d}\frac{\sigma_d^2}{\mu_d^2} \Longrightarrow d \to \infty \Rightarrow \sigma'^2 \to 0$$

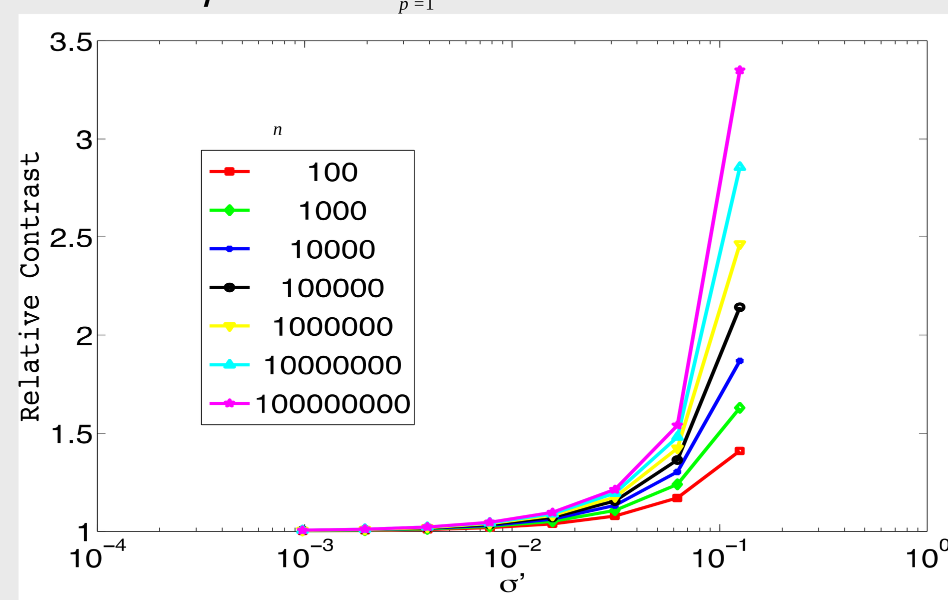
Distance to any point becomes roughly the same !

What Affect NN Search and How?

NN Search Difficulty — Relative Contrast C_r

$$C_r = \frac{D_{mean}}{D_{min}} \approx \frac{1}{[1 + \phi^{-1}(\frac{1}{n} + \phi(\frac{-1}{\sigma'}))\sigma']^{\frac{1}{p}}}$$

ϕ - standard Gaussian cdf



Larger σ' → higher C_r

Normalized Variance σ'

Suppose dimensions are i.i.d., and each dimension has a probability s of being non-zero

Probability of both x^j and q^j being non-zero = s^2

Probability of either x^j or q^j being non-zero = $2s(1-s)$

For non-zero entries, let $E\left[\left|x^j\right|^p\right] = m_p, E\left[\left|q^j - x^j\right|^p\right] = m_p'$

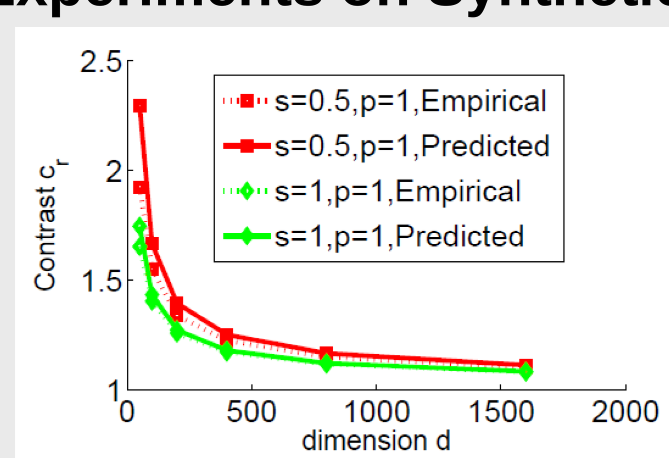
$$\mu_d = s^2 m_p' + 2s(1-s)m_p \qquad \sigma_d^2 = s^2 m_{2p}' + 2s(1-s)m_{2p} - \mu_d^2$$

$$\sigma' = \frac{1}{d^{1/2}}\sqrt{\frac{s[(m_{2p}' - 2m_{2p})s + 2m_{2p}]}{s^2[(m_p' - 2m_p)s + 2m_p]^2} - 1}$$

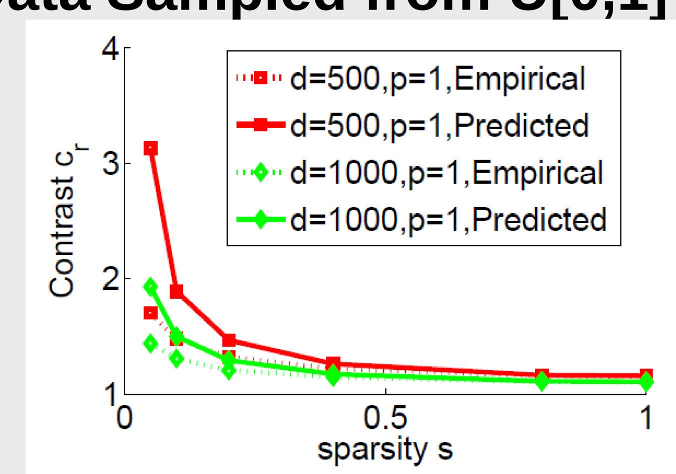
Data Properties:

dimension d, sparsity s, Lp distance p, database size n

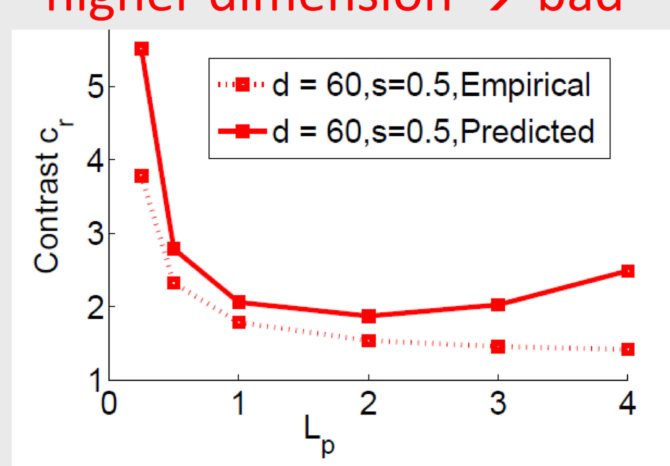
Experiments on Synthetic Data Sampled from U[0,1]



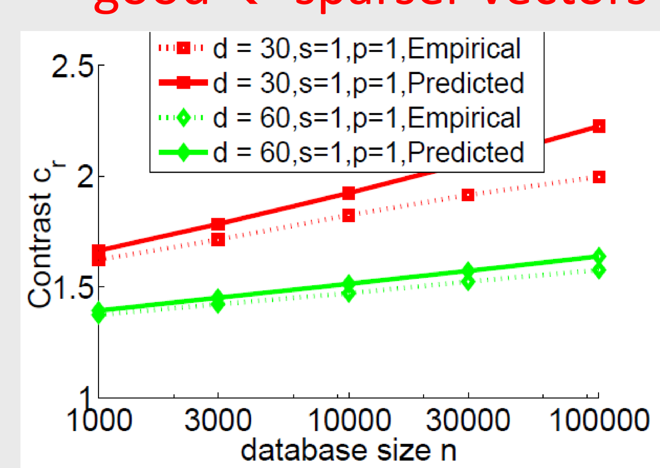
higher dimension → bad



good ← sparser vectors



good ← Lower p



Large database → good

Relative Contrast and LSH Complexity

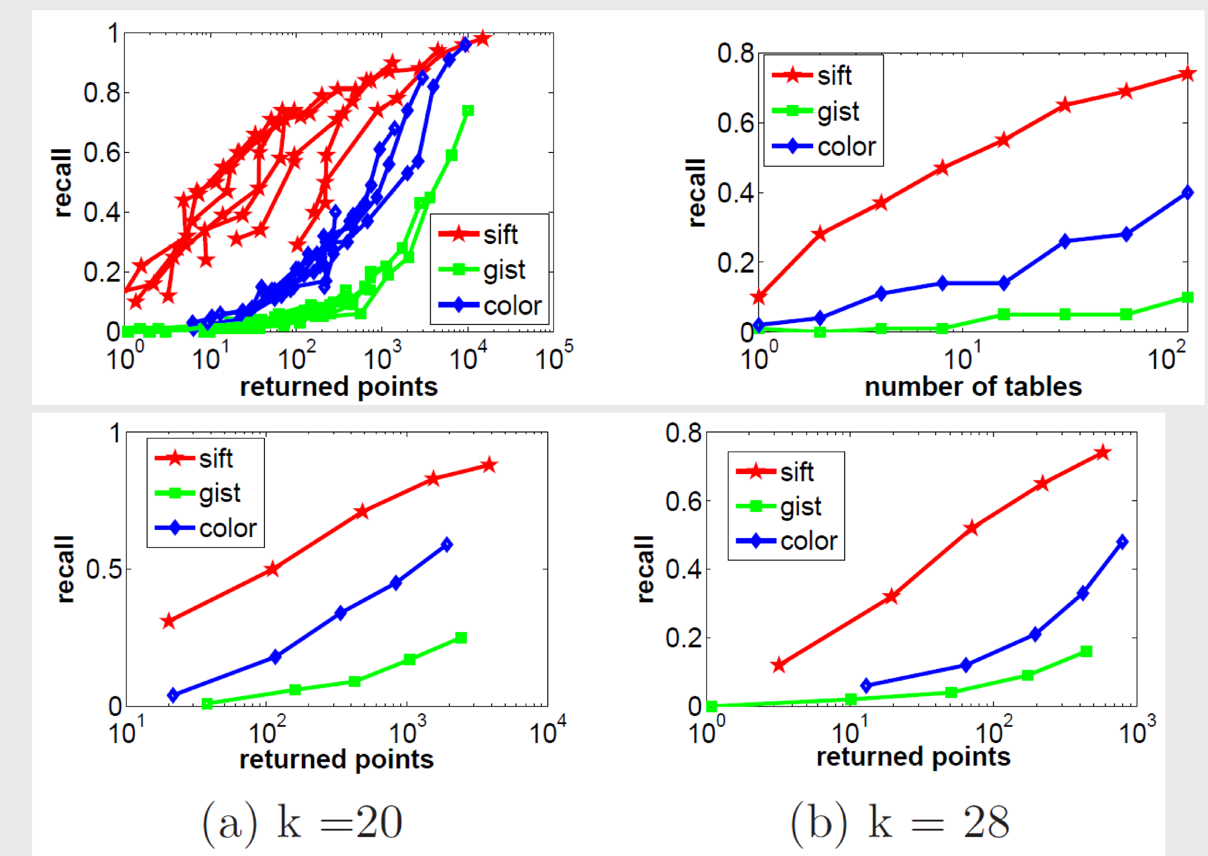
Theorem 3.1 *LSH can find the exact nearest neighbor with probability $1-\delta$ by returning $O(\log\frac{1}{\delta}n^{g(C_r)})$ candidate points, where $g(C_r)$ is a function monotonically decreasing with C_r.*

Corollary 3.2 *LSH can find the exact nearest neighbor with a probability at least $1-\delta$ with a time complexity $O(d\log\frac{1}{\delta}n^{g(C_r)}\log n)$ and space complexity $O(\log\frac{1}{\delta}n^{(1+g(C_r))} + nd)$. l, the number of hash tables needed, is $l = O(\log\frac{1}{\delta}n^{g(C_r)})$.*

Dataset	Dimensionality (d)	Sparsity (s)	Relative Contrast (C_r) for $p = 1$
SIFT	128	0.89	4.78
Gist	384	1.00	1.83
Color Hist	1382	0.027	3.19

LSH (with multiple hash table lookup)

LSH bits for hamming ranking



Relative Contrast and PCA Hashing

- **Linear Hashing**

$$h(x) = \mathrm{sgn}(w^T x + b),\ w \in \Re^d$$

– **Suppose** $b = E[x] = 0$

- **Want to find w such that relative contrast of projections is maximized**

Projected distance to nearest neighbor $(w^T q - w^T x_{nn})^2$

Expected distance to a random point $E_x(w^T q - w^T x)^2$

$$\hat{w} = \arg\max_w \frac{w^T E_q[\sum_i (q-x_i)(q-x_i)^T]w}{w^T E_q[(q-x_{nn})(q-x_{nn})^T]w} = \frac{w^T S_x w}{w^T S_{nn} w}$$

$$\hat{w} = \arg\max_w \frac{w^T S_x w}{w^T S_{nn} w} \Rightarrow eigenvec(S_{nn}^{-1} S_x)$$

If distribution of nearest neighbors is isotropic

$$S_{nn} \approx \alpha I$$

If queries have the same distribution as data, $q \sim x$

$$S_x \approx \beta \Sigma_x$$

$$\hat{w} = eigenvec(\Sigma_x)$$

PCA-directions !

Recall of 1-NN with PCA hashing and Modified PCA hashing (MRC) for Hamming Ranking