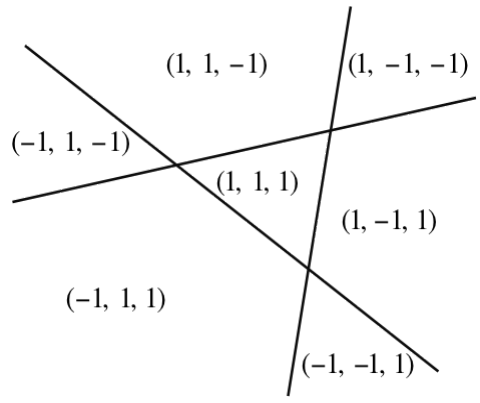
How to select hashing bits with a simple measurement? An alternative greedy approach

Xiping Fu

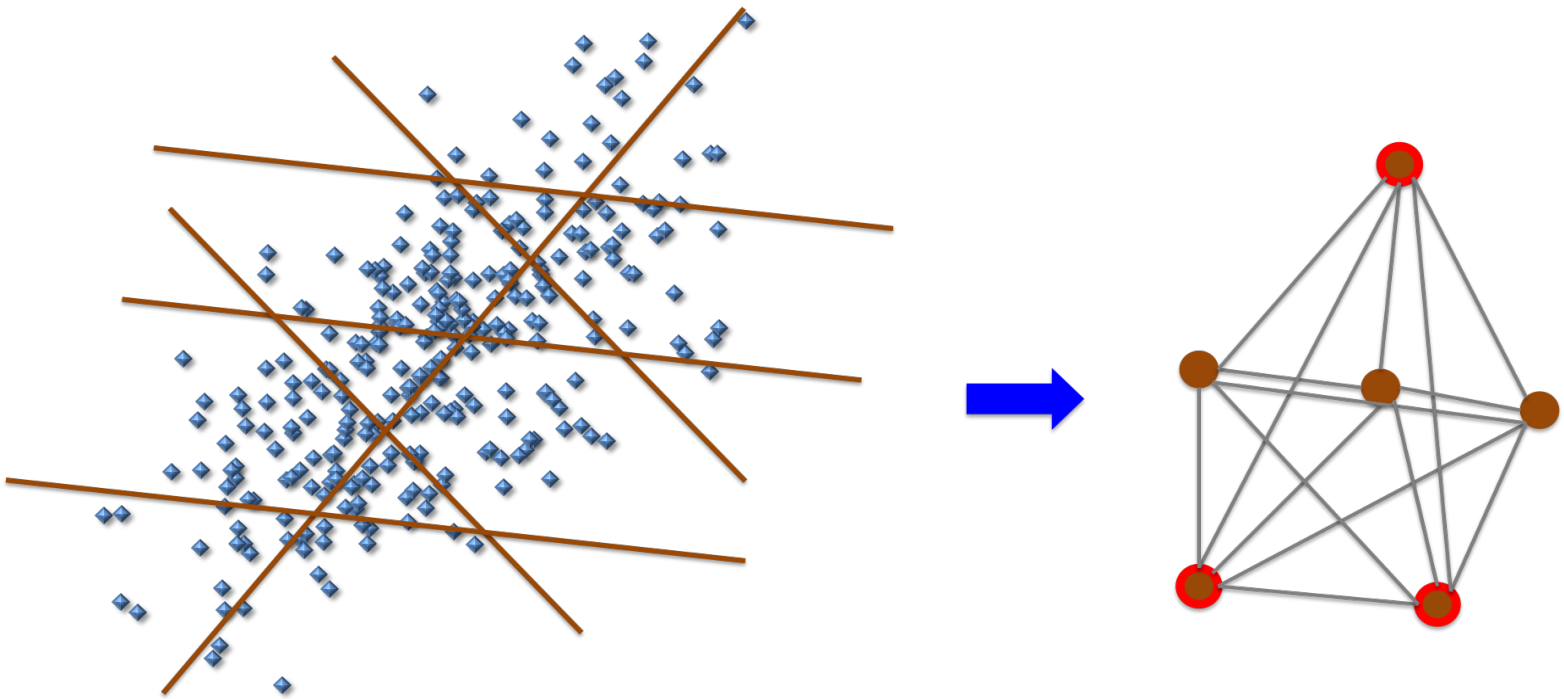
August 24, 2015

Hashing algorithms for KNN problems

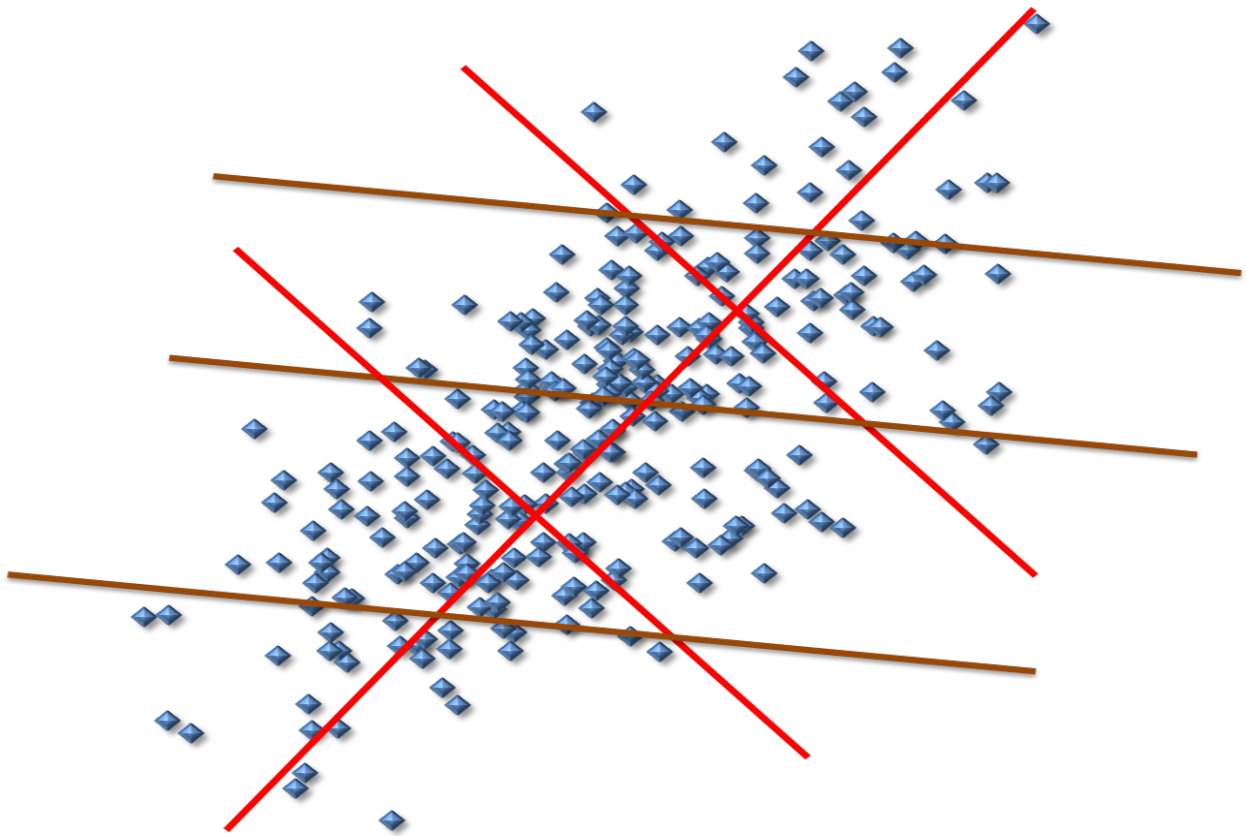
- Locality Sensitive Hashing (LSH)
  Preserve the similarity if enough bits are used.
- Spectral Hashing
  Design compact bits which preserves the local similarity.
- Iterative Quantization (ITQ)
  Minimize the quantization error.
- . . .



Previous bits selection methods



Motivation



Measuring the quality of a bit set

For each query data point, its nearest n data points in sense of the Hamming distance are retrived. Recall@n is the percentage of true nearest neighbor points in the retrieved data set, i.e.:

$$\text{Recall@}n = \frac{\#\text{retrieved true nearest neighbor points}}{\#\text{true nearest neighbor points}}$$

Then the m-Recall is calculated as:

$$\text{m-Recall} = \frac{\sum_{n=1}^{K} \text{Recall@}n}{K}$$

here K is the maximum retrieved nearest points in Hamming space and Recall@n is the average recall of retrieving n nearest neighbor points.

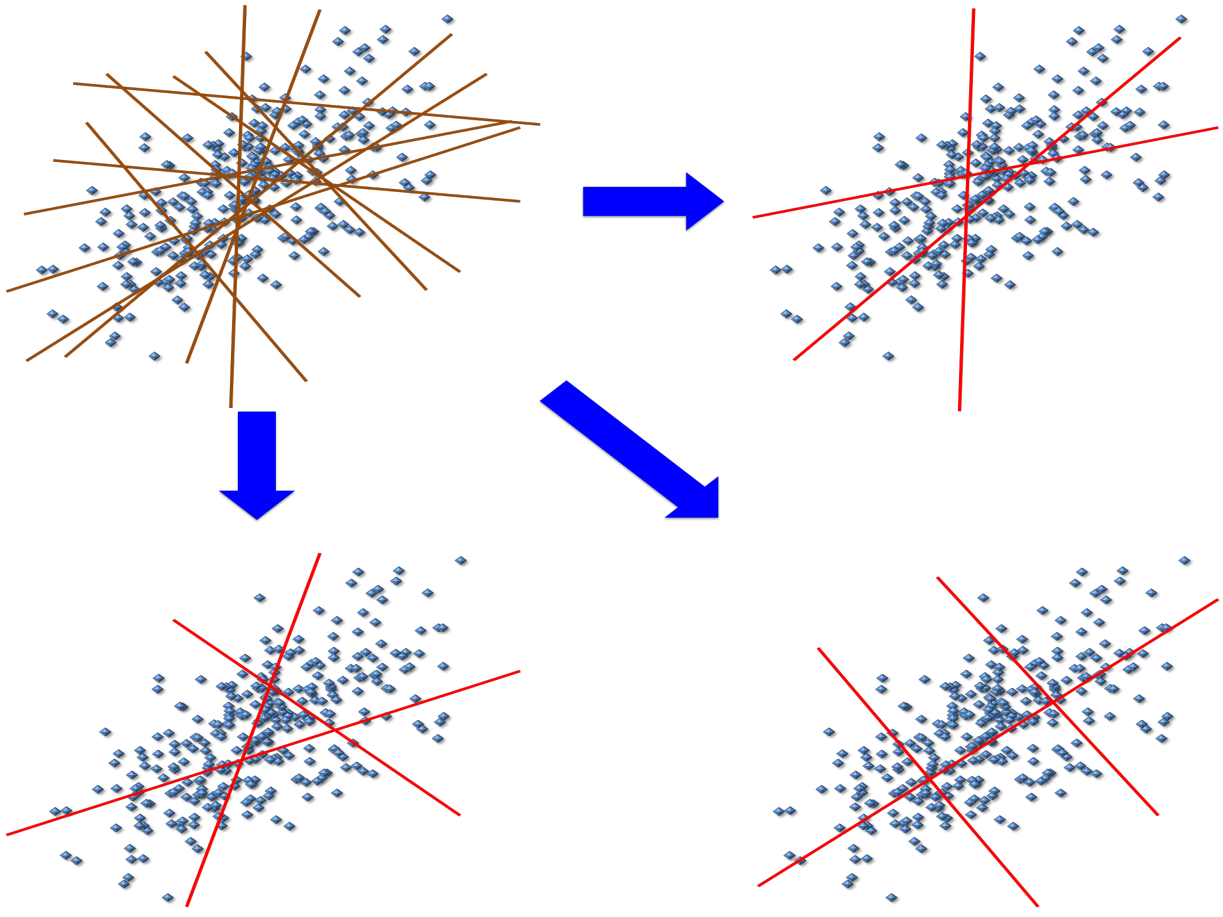
Bits selection with alternative greedy method I

- The task is to select a set $\mathcal{S}$ with M bits from the pool $\mathcal{P}$ with size N.
- $\mathcal{S}$ is initialed with the bits randomly selected from $\mathcal{P}$.
- For updating the i-th bit, the position is updated with the bit which has the highest m-Recall score.

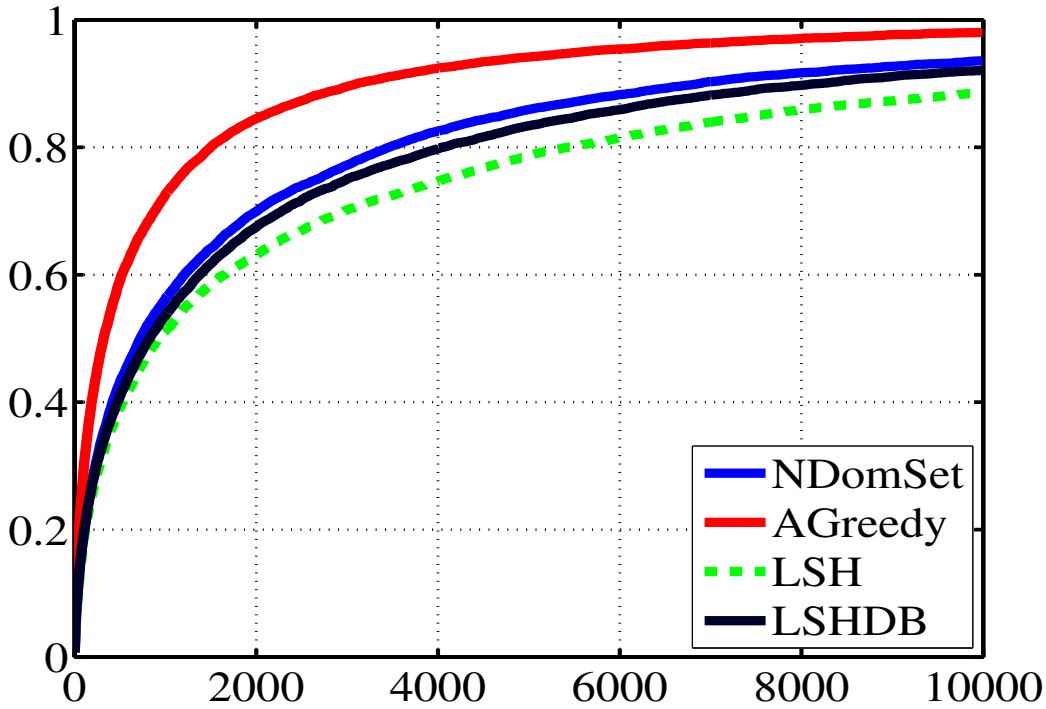
Bits selection with alternative greedy method II

- For updating the i-th bit, the hamming distance between the remaining $M - 1$ bits can be precomputed.
- Measuring each candidate bit parallelly.
- The optimization method is guaranteed to obtain a local optimal solution.

Why it works?



Results



- Dataset:
  NUSWIDE, 128D, training (10,000), query (10,000), base datasets (249,648), Pool (500).
- Retrieving task: 32 bits, 1-NN.
- Performance metric:

$$Recall@N = \frac{\text{The number of retrieved true nearest neighbor points}}{\text{The number of retrieved data points}}$$

Thank you for your time!