CLSH: Cluster-based Locality-Sensitive Hashing


Xiangyang Xu


Tongwei Ren


Gangshan Wu

Multimedia Computing Group, State Key Laboratory for Novel Software Technology, Nanjing University

xiangyang.xu@smail.nju.edu.cn

Outline

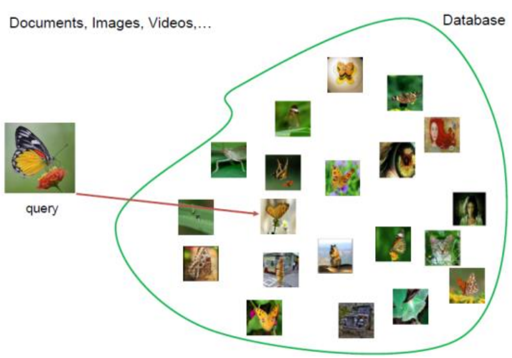
- Background
- Approach
- Experiment
- Conclusion

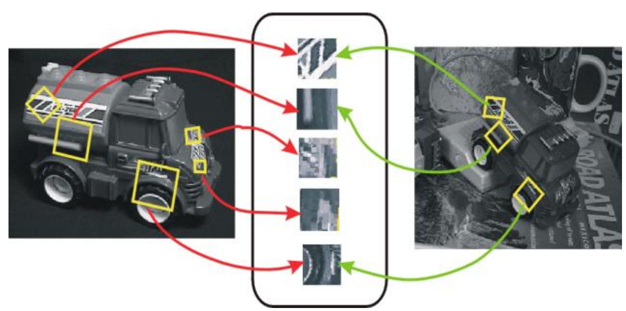
Outline

- Background
- Approach
- Experiment
- Conclusion

Nearest neighbor search

- Search over millions, even billions of data
  - Images, local features, other media objects, …
- Applications
  - Image retrieval, computer vision, machine learning, …





Challenges

- Query precision and recall
  - Basic requirements in nearest neighbor search

Effectiveness

- Query speed
  - For high-dimensional spaces, there is no any generic exact algorithm that is faster than linear search [M. Muja, 2013]
  - *O(n)* complexity is prohibitive

Efficiency

- Memory cost
  - Increase in number of dimensions leads to rapid increase in volume

Scalability

M. Marius. "Scalable nearest neighbour methods for high dimensional data." (2013).

Challenges

- Query precision and recall
  - Basic requirements in nearest neighbor search

Effectiveness

- Query speed
  - For high-dimensional spaces, there is no any generic exact algorithm that is faster than linear search [M. Muja, 2013]
  - *O(n)* complexity is prohibitive

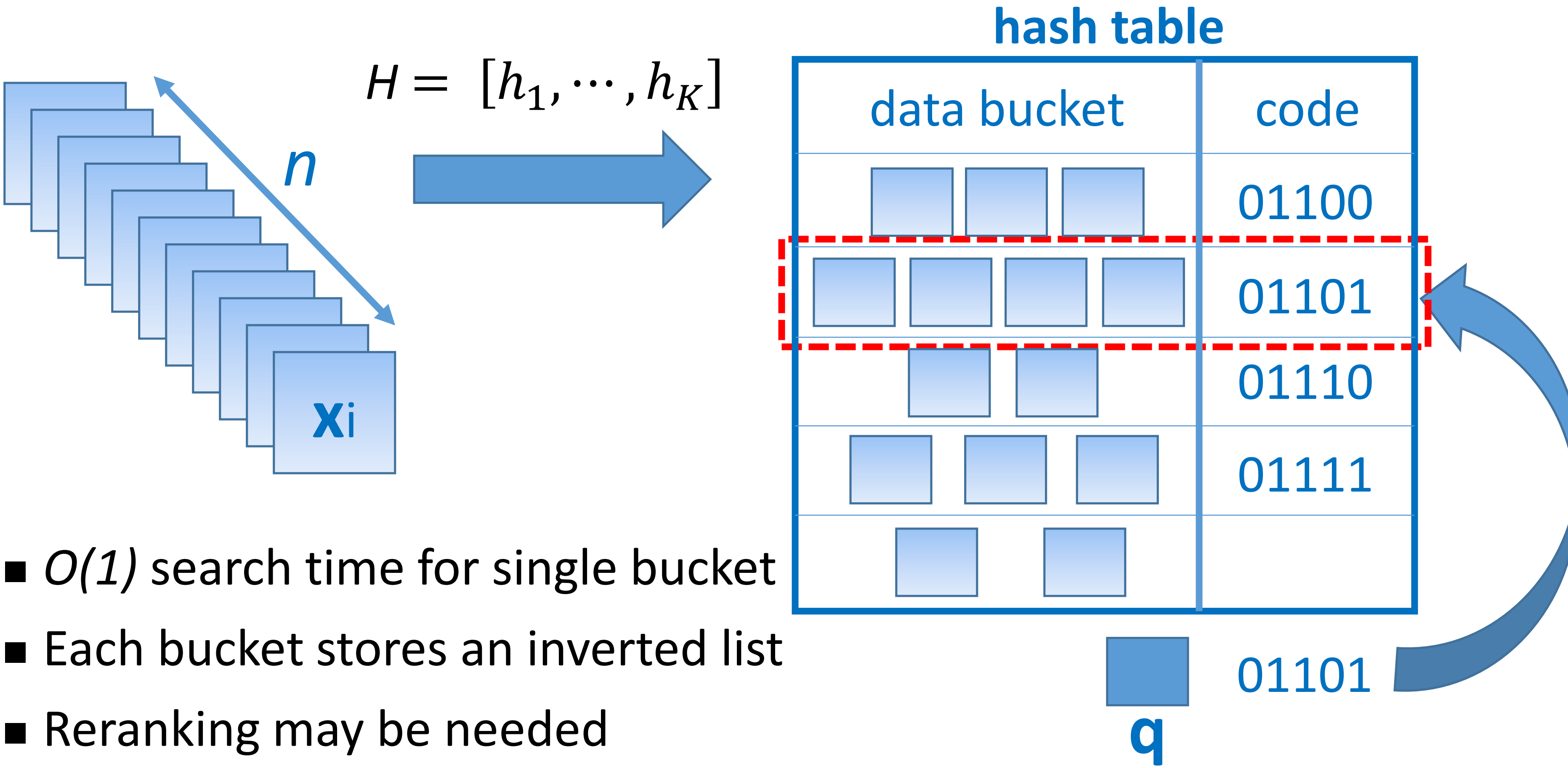
Efficiency

- Memory cost
  - Increase in number of dimensions leads to rapid increase in volume

Scalability

M. Marius. "Scalable nearest neighbour methods for high dimensional data." (2013).

Hashing-based methods



- *O(1)* search time for single bucket
- Each bucket stores an inverted list
- Reranking may be needed
- LSH, spectral hashing, semi-supervised hashing, weakly-supervised hashing and kernelized LSH, …

Motivation and Contribution

- Cluster-based
  - Clustering algorithm
  - Index is carried out on a distributed cluster
- Centralized settings → distributed settings
  - CLSH can cope with larger scale feature dataset
    - **Clustering and hashing**
  - The generated clusters can guide feature dataset automatic mappings to a distributed cluster
    - **One node cover one cluster**
  - Search time is significantly reduced
    - **Parallel searching on multiple computing nodes**

Motivation and Contribution

- Cluster-based
  - Clustering algorithm
  - Index is carried out on a distributed cluster
- Centralized settings → distributed settings
  - CLSH can cope with larger scale feature dataset

Efficiency & Scalability

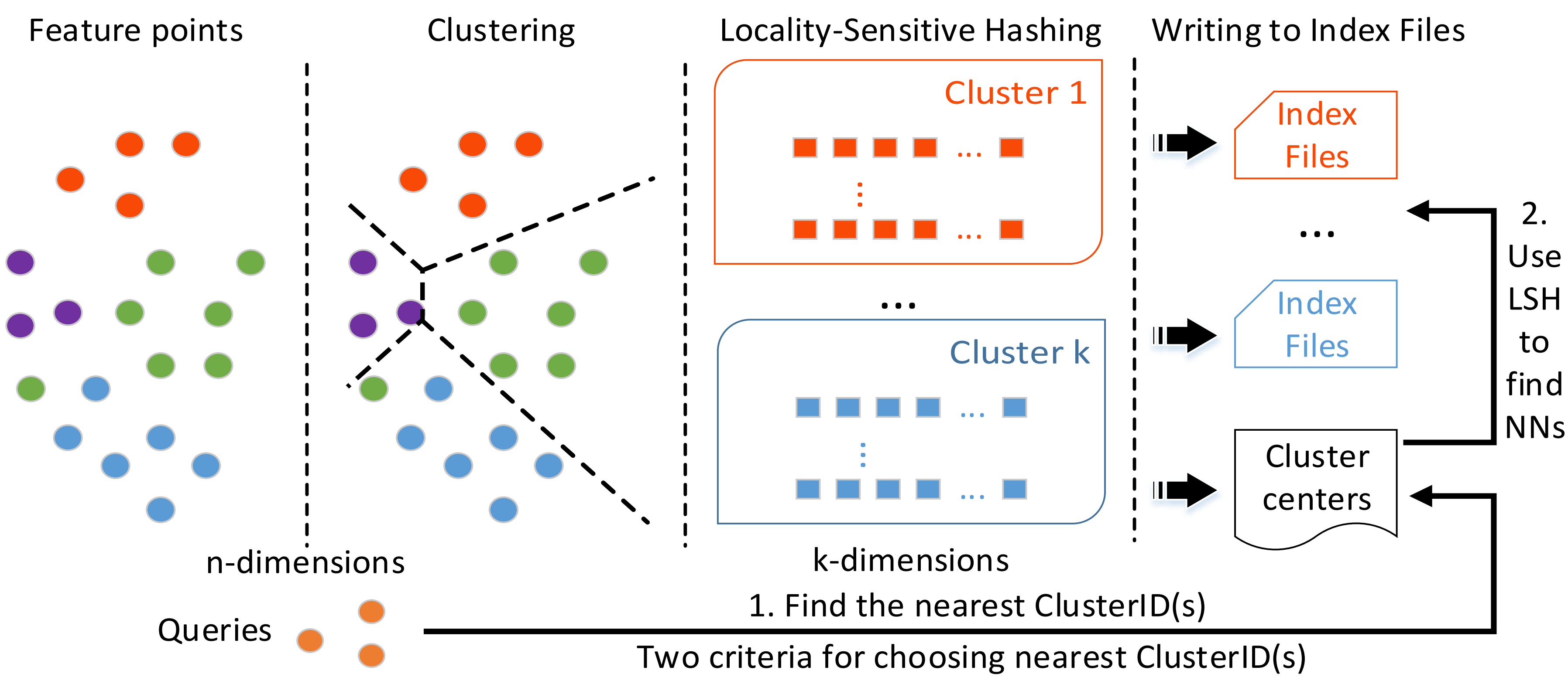
automatic mappings to a distributed cluster
    - **One node cover one cluster**
  - Search time is significantly reduced
    - **Parallel searching on multiple computing nodes**

Outline

- Background
- Approach
- Experiment
- Conclusion

Approach

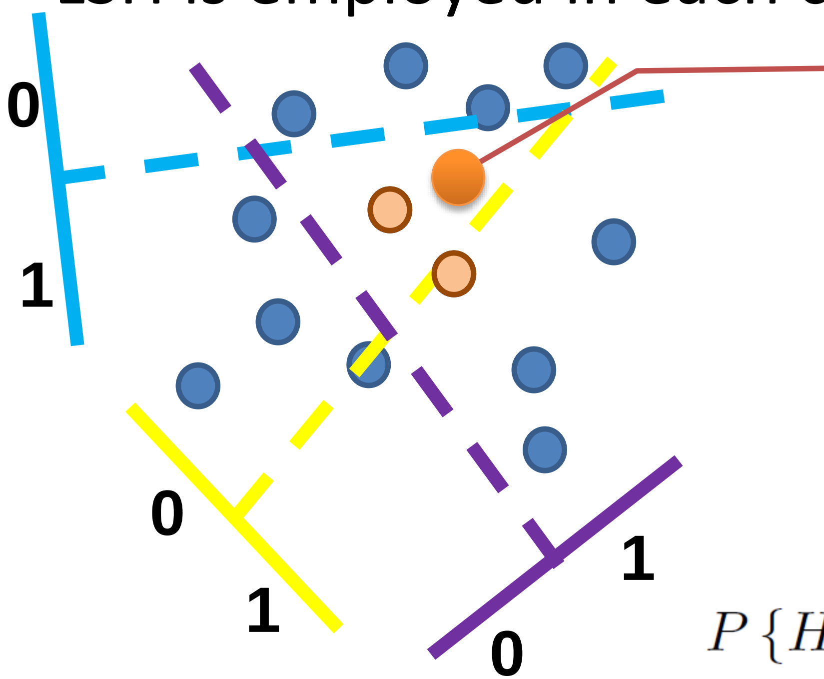
- Index construction
- Nearest neighbor searching



Indexing construction

- Clustering the feature dataset
  - k-means
- LSH is employed in each cluster

[P. Indyk 1998, M. Datar 2004]



110 **Index by compact code**

hash function

$$h(\mathbf{x}) = \mathrm{sgn}(\mathbf{w}^\top \mathbf{x} + b)$$

random

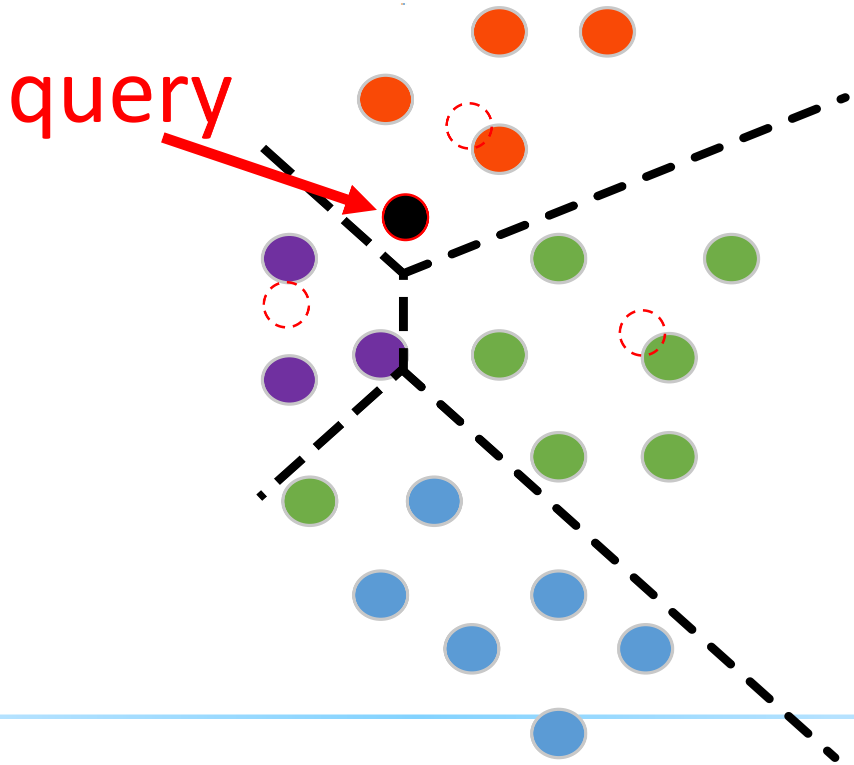
$$P\{H(\mathbf{x}) = H(\mathbf{y})\} = l \cdot \left[1 - \frac{cos^{-1}\mathbf{x}^\top \mathbf{y}}{\pi}\right]^K$$

Prob(hash code collision) is proportional to data similarity

l: # hash tables, *K*: hash bits per table

Nearest neighbor searching

- Query near the cluster boundary
  - Search fixed number s clusters
  - **Search the clusters:** $\frac{d_i}{\min\{d_i\}} \leq T(i = 1, \cdots, k)$



Outline

- Background
- Approach
- Experiment
- Conclusion

Experiments

- Experiment settings
  - Dataset
    - INRIA BIGANN (10K 128-d SIFT, 1M SIFT, 1M 960-d GIST)
- LSH is a filter-and-refine framework, only recall is employed for measurement

Results

Table 1: Comparison on Recall

Dataset		SIFT10K	SIFT1M	GIST1M
E2LSH		0.9647	0.9494	0.9680
CLSH	$s = 1$	0.8704	0.8926	0.7732
	$s = 2$	0.9667	0.9494	0.9514
	$s = 3$	0.9741	0.9494	0.9647
	$T = 1.1$	0.9518	0.9319	0.8953
	$T = 1.2$	0.9741	0.9494	0.9640
	$T = 1.3$	0.9741	0.9494	0.9647

Table 2: Comparison on the detailed distance evaluation times

Dataset		SIFT10K	SIFT1M	GIST1M
E2LSH		142.6	13,435.3	121,871
CLSH	$s = 1$	95.03	9,854.27	53,021.7
	$s = 2$	124.64	13,318.5	91,421.2
	$s = 3$	134.5	14,639.4	106,805
	$T = 1.1$	108.17	11,078.8	75,891.2
	$T = 1.2$	119.32	12,753.2	93,990
	$T = 1.3$	128.46	13,467	107,738

Results (cntd.)

- Search time in our settings
  - 6 computing nodes (64-bit 2.00GHz, 8GB RAM each)

Table 3: Comparison on total search time (s)

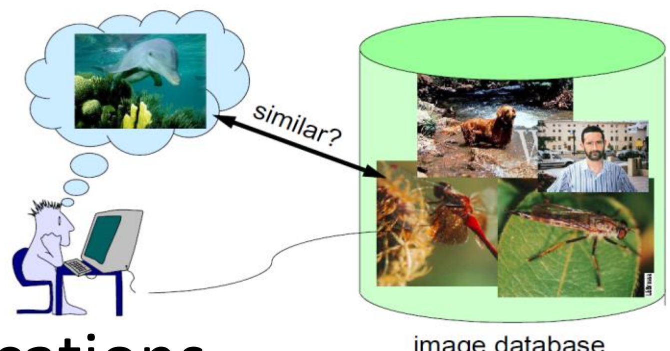
Dataset	E2LSH	CLSH						$\text{Max}\{T_{c_i}\}$
		$s=1$	$s=2$	$s=3$	$T=1.1$	$T=1.2$	$T=1.3$	
SIFT10K	0.00031	0.00021	0.00022	0.00024	0.00022	0.00024	0.00025	0.00022
SIFT1M	0.01531	0.00813	0.00907	0.00994	0.00915	0.00983	0.01011	0.00813
GIST1M	0.59721	0.25116	0.25832	0.26014	0.25883	0.26001	0.26797	0.25271

Outline

- Background
- Approach
- Experiment
- Conclusion

Conclusion

- A distributed scalable framework for large-scale high-dimensional datasets indexing and searching
- Clustering is applied and the generated clusters are treated as a guideline to automatically deliver the feature dataset to a distributed cluster
- The search time is significantly reduced in CLSH framework



- Data-adaptive hashing function
- Extend our work to further applications

References

- P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In STOC, pages 604–613. ACM, 1998.
- M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In SoCG, pages 253–262. ACM, 2004.
- B. J. Frey and D. Dueck. Clustering by passing messages between data points. Science, 315:972–976, 2007.
- J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In CVPR, pages 1–8. IEEE, 2007.
- J. Wang, S. Kumar, and S.-F. Chang. Semi-supervised hashing for scalable image retrieval. In CVPR, pages 3424–3431. IEEE, 2010.
- Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In NIPS, pages 1753–1760. MIT Press, 2008.

Thank you!