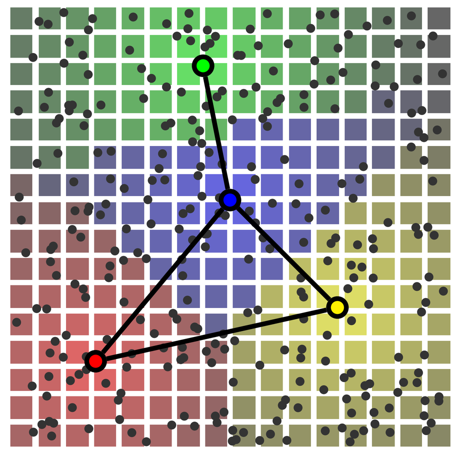
Quantize and Conquer: A dimensionality-recursive solution to clustering, vector quantization, and image retrieval

Yannis Avrithis, NTUA

Motivation

- Connection between *clustering* and *approximate nearest neighbor* (ANN) search
  - *approximate k-means* (AKM) [1]: use ANN search to accelerate assignment step
  - *product quantization* (PQ) [2]: use k-means on subspaces to accelerate ANN search
  - *inverted multi-index* [3]: exhaustively search on subspaces before searching on entire space
- What is the actual connection under subspace decomposition? Is there something missing?
- Can we use recursion to solve both problems at the same time?

Problem



- Given n points in d dimensions, quantize to k centroids under minimal distortion, with $n > 10^6, d > 10^2, k > 10^3$
- k-means assignment step is the bottleneck
  - exhaustive search: $O(nk)$ time; ANN search (AKM): e.g., $O(n \log k)$
  - n nearest neighbor queries over the *same set* of k centroids
  - so why not lookup on precomputed distance maps and Voronoi cells?
  - $O(n)$ time, but $O(2^d)$ space: fine e.g. for $d = 2$
- But what if $d > 10$? Is then a lookup-based solution possible?
- Our *dimensionality-recursive clustering* (DRC) takes $O(k^3)$ time to pre-process and $O(n)$ time to assign, at $O(k^2)$ space

DRC Base case: one dimension

Given

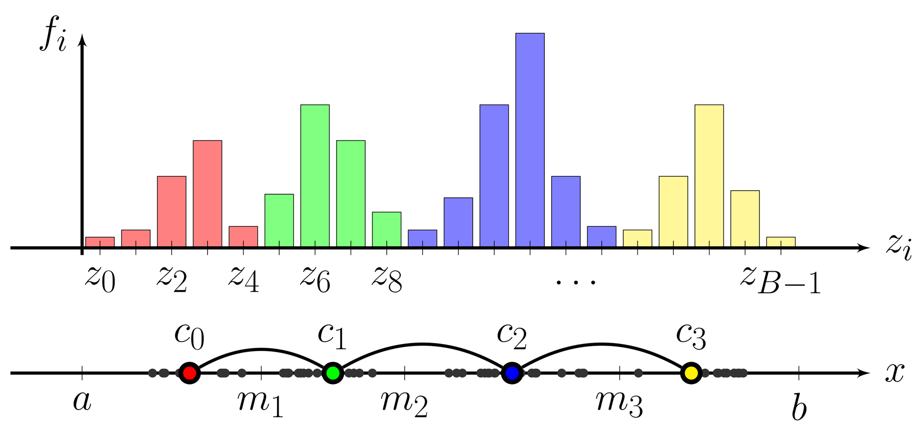
- set X of N data points on interval $I = [a, b)$ of $\mathbb{R}$
- target number $K > 1$ of centroids

Representation

- partition I into $B \gg K$ subintervals (bins) of length $\ell = (b-a)/B$
- let $Z = \{z_0, \ldots, z_{B-1}\}$ be the midpoints of subintervals
- allocate $x \in X$ to bin $r(x) = \lfloor (x-a)/\ell \rfloor \in \{0, \ldots, B-1\}$
- quantize points via $h : I \to Z$ with $x \mapsto h(x) = z_{r(x)} = a + \ell r(x) + \ell/2$

Initialization

- let $X_i = \{x \in X : r(x) = i\}$ be the set of points allocated to bin i
- measure discrete distribution f by normalized histogram frequency $f_i = |X_i|/N$
- centroids $C = \{c_0, \ldots, c_{K-1}\}$: K samples out of Z with replacement, according to f



Quantizer

- ideal: $q : I \to C$ with $x \mapsto q(x) = \arg\min_{c \in C} \|x - c\|$
- approximation: restriction $q^* : Z \to C$, *i.e.*, compute $q(z)$ and store as $q^*[z]$ for all $z \in Z$.

Assignment step

- let m_k be the midpoint of $[c_{k-1}, c_k)$ for $k = 1, \ldots, K-1$; $m_0 = a$, $m_K = b$
- then Voronoi cell $V_k = \{z \in Z : q(z) = c_k\}$ found as $Z \cap [m_k, m_{k+1})$ for all $c_k \in C$
- assign $q^*[z] \leftarrow c_k$ for all $z \in V_k$

Update step

- weighted averaging over Voronoi cells: $c_k \leftarrow \sum_{i : z_i \in V_k} f_i z_i$ for all $c_k \in C$

At termination

- approximate $q(x) \simeq q^*[h(x)] \in C$ for all $x \in X$
- construct graph $G = \{C, E\}$ with edges $E = \{(c_{k-1}, c_k) : k = 1, \ldots, K-1\}$ between successive centroids as a neighborhood system over I

DRC Recursion: from d to $2d$ dimensions
(or: learning a joint distribution from two marginal ones)

Subspace decomposition

- decompose $2d$-dimensional space S into product $S^L \times S^R$ of d-dimensional subspaces S^L, S^R
- write $x \in S$ as $x = (x^L, x^R)$ with projections $x^L \in S^L, x^R \in S^R$

Given

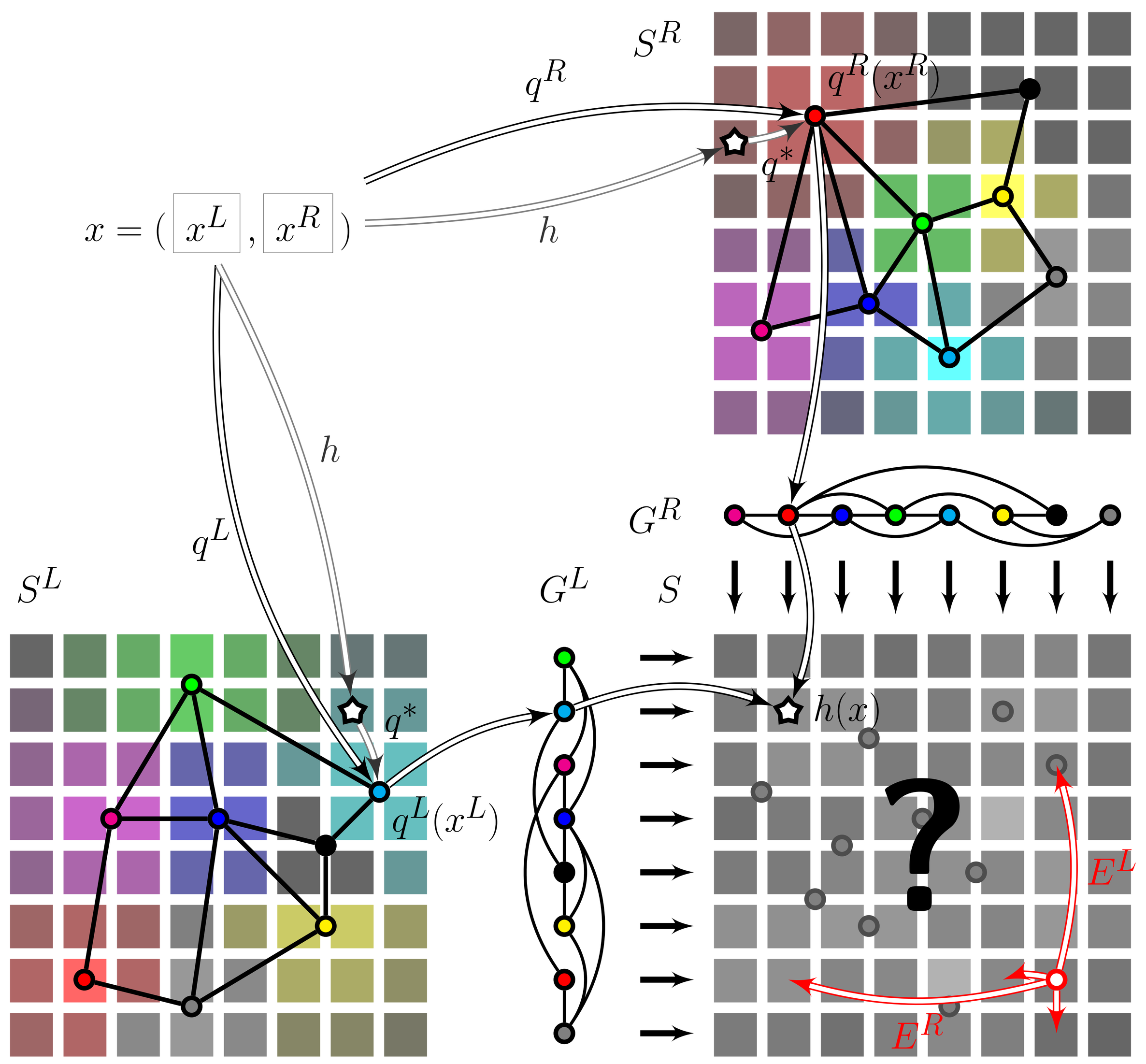
- set X of N data points on interval $I = I^L \times I^R$ of S
- target number $K > 1$ of centroids
- sets of projections X^L, X^R clustered into C^L, C^R, each of J centroids
- each projection x^L (x^R) quantized to $q^L(x^L) \in C^L$ ($q^R(x^R) \in C^R$)
- graphs $G^L = \{C^L, E^L\}$, $G^R = \{C^R, E^R\}$ representing neighborhood systems over I^L, I^R

Representation

- let $Z = C^L \times C^R$ be a grid of $B = J \times J$ points in S
- write $Z = \{z_0, \ldots, z_{B-1}\}$: again, a discrete representation of I
- quantize points via $h : I \to Z$ with $x \mapsto h(x) = (q^L(x^L), q^R(x^R))$

Initialization

- let $X_i = \{x \in X : h(x) = z_i\}$ be the set of points allocated to bin i
- measure f with $f_i = |X_i|/N$ and sample $C = \{c_0, \ldots, c_{K-1}\}$ as in one dimension



Clustering

- assignment: compute $q(z)$ and store as $q^*[z]$ for all $z \in Z$: *product propagation*, $O(K^3)$
- update: exactly as in one dimension

At termination

- quantize centroids to nearest points on grid Z as $c_k \leftarrow h(c_k)$ for $c_k \in C$
- approximate $q(x) \simeq q^*[h(x)] \in C$ for all $x \in X$
- compute graph $G = \{C, E\}$ once at final assignment step, as by-product of propagation

References

[1] J. Philbin *et al.* Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.
[2] H. Jégou *et al.* Product quantization for nearest neighbor search. *PAMI* 33(1), 2011.
[3] A. Babenko and V. Lempitsky. The inverted multi-index. In *CVPR*, 2012.

Dim.-recursive quantization (DRQ)

Approximate quantization

- recursively compute $q(x)$ by delegating $q^L(x), q^R(x)$ if $d > 1$:

$$q(x) \simeq \begin{cases} q^*[a + \ell r(x) + \ell/2], & d = 1 \\ q^*[q^L(x^L), q^R(x^R)], & d > 1 \end{cases}$$

- time complexity when $D = 2^P$: $O(D)$
  - tree structure with D leaves and $D-1$ internal nodes
  - hence, D scalar quantizations and $D-1$ lookups
- not precise enough for NN search, but fine for k-means assignment

Exact quantization

- recursively compute squared Euclidean distance to all centroids
- $d = 1$: compute $\delta(x, c) = (x-c)^2$ for all $c \in C$.
- $d > 1$:
  - delegate $\delta^L(x^L, z^L)$, $\delta^R(x^R, z^R)$, for all $z \in Z$
  - let $\delta(x, z) = \delta^L(x^L, z^L) + \delta^R(x^R, z^R)$ for any $x \in I, z \in Z$
  - minimize $q(x) = \arg\min_{c \in C} \delta(x, c)$
- exact because centroids are stored for $d = 1$ and quantized on grid for $d > 1$
- time complexity with $D = 2^P$ (tree of height P), K_p centroids at 2^p dimensions (level p) and $\mathcal{K} = \{K_0, \ldots, K_P\}$
  - recursive: $O(\phi(\mathcal{K})) = O(K \log D)$ where $\phi(\mathcal{K}) = \sum_{p=0}^{P} 2^{P-p} K_p$
  - naïve: $O(K_P 2^P) = O(KD)$

Product propagation

function $(q^*, E) \leftarrow \text{PP}(C, Z, h, \delta; E^L, E^R, \tau)$
  $E \leftarrow \emptyset$; initialize queue Q
  for $z \in Z$ **do** *state*$[z] \leftarrow$ ALIVE
  for $c \in C$ **do** PUSH$(c, h(c))$
  while $\neg Q$.EMPTY() **do**
    $z \leftarrow Q$.EXTRACT-MIN()
    state$[z] \leftarrow$ FAR; $c \leftarrow q^*[z]$
    for $y \in E^L(z^L)$ **do** SCAN$(c, (y, z^R))$
    for $y \in E^R(z^R)$ **do** SCAN$(c, (z^L, y))$
  return (q^*, E)

function SCAN(c, z)
  if *state*$[z] =$ ALIVE **then** PUSH(c, z)
  if *state*$[z] =$ CLOSE **then** RELAX(c, z)
  if *state*$[z] =$ FAR **then** JOIN(c, z)

function PUSH(c, z)
  dist$[z] \leftarrow \delta(c, z)$; $q^*[z] \leftarrow c$
  Q.INSERT(z); *state*$[z] \leftarrow$ CLOSE

function RELAX(c, z)
  $d \leftarrow \delta(c, z)$
  if $d <$ *dist*$[z]$ **then**
    dist$[z] \leftarrow d$; $q^*[z] \leftarrow c$
    Q.DECREASE-KEY(z, d)

function JOIN(c, z) ▷ only at termination
  if $\delta(c, z) +$ *dist*$[z] < \tau$ **then**
    $E \leftarrow E \cup (c, q^*[z])$

Experiments

Clustering

4 codebooks at $D = 32$ dimensions each on $N = 12.5$M 128-dimensional SIFT descriptors of Oxford 5K

K	$\log K_p$ ($d = 2^p$)						time (m)
	1	2	4	8	16	32	
16K	6	7	8	9	11	14	129.96
8K	6	7	8	9	11	13	119.43
4K	6	7	8	9	10	12	20.07
2K	5	6	7	8	9	11	2.792
1K	5	6	7	8	9	10	2.608
512	4	5	6	7	8	9	0.866
4K	AKM [1]						504.2

Vector quantization

averaged over the $N = 75$K SIFT descriptors of the 55 cropped query images of Oxford 5K

K	16K	8K	4K	2K	1K	512
Approximate (μs)	0.95	0.83	0.80	0.73	0.80	0.90
Exact (ms)	1.19	0.79	0.51	0.26	0.21	0.11

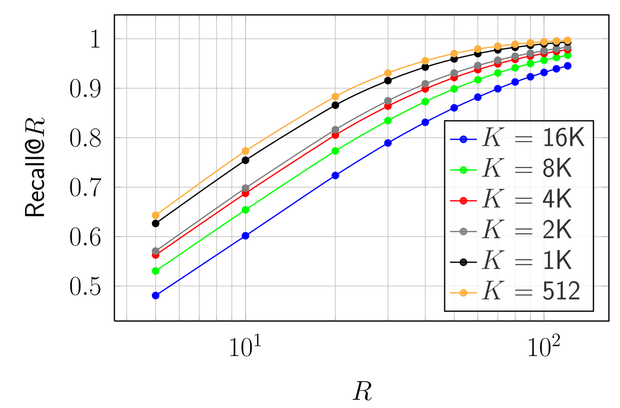


Image retrieval

fourth-order multi-index [3] with 4K sub-codebooks, partially inverted at 24bit/point, MA $k = 90$

Training set	Oxford 5K / other [*]				Paris 6K / other [*]		K	MA	Other
Test set	Ox5K	Ox105K	Pa6K	Pa106K	Ox5K	Ox105K			
This work	0.716	**0.657**	**0.696**	**0.584**	**0.703**	**0.640**	$4K^4$	✓	
Perdoch *et al.* 2009	**0.717**	0.568	—	—	0.558	0.423	1M		
Arandjelovic *et al.* 2012	0.683	0.581	—	—	—	—	1M		
Shen *et al.* 2012	0.649	0.568	—	—	—	—	1M		
Philbin *et al.* 2008	0.614	0.498	—	—	0.403	0.290	1M		
Philbin *et al.* 2008	0.673	0.534	—	—	0.493	0.343	1M	✓	
Philbin *et al.* 2007	0.618	0.490	—	—	—	—	1M		
Jegou *et al.* 2010	—	—	—	—	0.615	0.516	200K	✓	HE, WGC
Jegou *et al.* 2009	—	—	—	—	0.647	—	20K	✓	HE, WGC
Mikulik *et al.* 2012	—	—	0.625*	0.533*	0.618*	0.554*	16M	✓	
Mikulik *et al.* 2012	—	—	**0.749***	**0.675***	**0.742***	**0.674***	16M	*	Learning