Spring25 CS598YP

8.2 Product Quantization

Yongjoo Park

University of Illinois at Urbana-Champaign

Our P2 will compare LSH and HNSW

Jan 30, 2025	4	Cloud: SPANStore	P1: Presto
Feb 4, 2025	5	Cloud: FlexPushdown	
Feb 6, 2025	6	Cloud: TELEPORT	
Feb 11, 2025	7	VectorDB: LSH	
Feb 13, 2025	8	VectorDB: Product Quantize	
Feb 18, 2025	9	VectorDB: HNSW	
Feb 20, 2025	10	VectorDB: DiskANN	
Feb 25, 2025	11	VectorDB: ACORN	P2: HNSW *(Feb 24: P1 due)*
Feb 27, 2025	12	Tuning: DB-BERT	
Mar 4, 2025	13	Tuning: AirIndex	
Mar 6, 2025	14	Tuning: Bao	
Mar 11, 2025	15	Tuning: QueryFormer	

Recap: Locality Sensitive Hashing

Embedding using llama

Doc ID: 29eaa338-e12e-4a03-838c-8d9e51faeec2

Text: Barack Hussein Obama II[a] (born August 4, 1961) is an American politician and lawyer who served as the 44th president of the United States from 2009 to 2017. A member of the Democratic Party, he was the first African-American president in U.S. history. Obama previously served as a U.S. senator representing Illinois from 2005 to 2008 and as an I...

Embedding dimension: 768

Overview; LSH

- given a query *q* (or not), how do we find similar items from a large search set quickly?

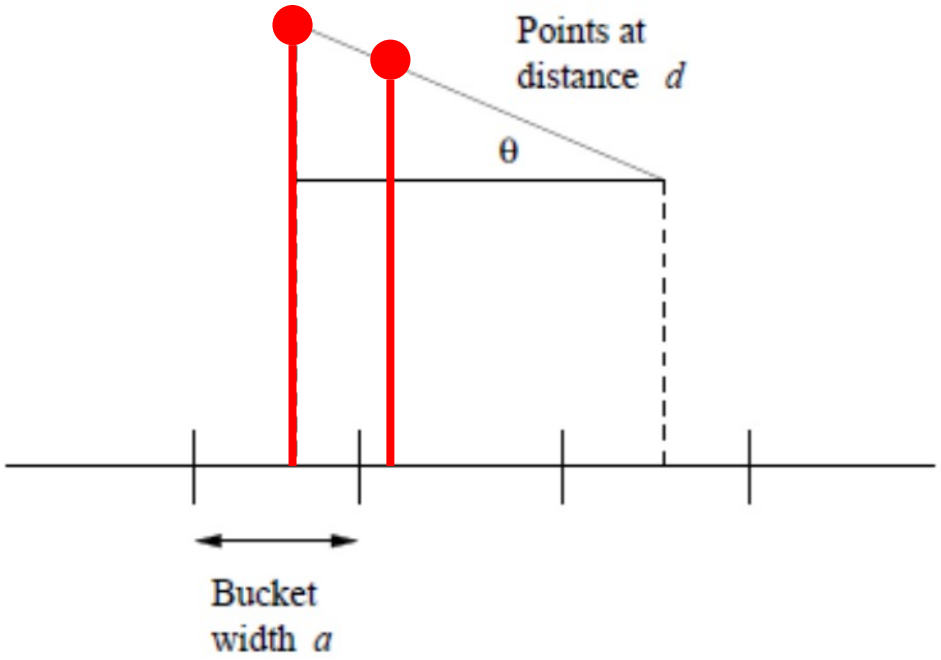
- define a measure of similarity for the items, then hash them into buckets using the measure.
  - Items which are similar will be in the same bucket.
- then when given a query *q*, we hash it and return items in the same bucket.

DEFINITION 1. *A family* $\mathcal{H} = \{h : S \to U\}$ *is called* (r_1, r_2, p_1, p_2)-sensitive *for* D *if for any* $v, q \in S$

- *if* $v \in B(q, r_1)$ *then* $\Pr_{\mathcal{H}}[h(q) = h(v)] \geq p_1$,
- *if* $v \notin B(q, r_2)$ *then* $\Pr_{\mathcal{H}}[h(q) = h(v)] \leq p_2$.

- use a randomly chosen line in 2-space (for each hash fn)
- select a constant *a* to divide line into equal width segments
- points projected onto the line, buckets are the segments
- *(a/2, 2a, 1/2, 1/3)*-sensitive family

n-dimensional random vector



p-Stable Distribution Scheme

- locality-sensitive families for l_p norm using *p*-stable distribution
  - eg. Gaussian distribution is 2-stable

projection onto a random vector

- distribution is stable if
  - $\sum_i v_i X_i$ has same distribution as $(\sum_i |v_i|^p)^{1/p} X$

original distance

- so with v & X as vectors the dot product estimates the l_p norm

Background: Spectral Hashing

$$minimize: \sum_{ij} W_{ij} \|y_i - y_j\|^2$$

$$subject\ to: y_i \in \{-1, 1\}^k$$

$$\sum_i y_i = 0$$

$$\frac{1}{n} \sum_i y_i y_i^T = I$$

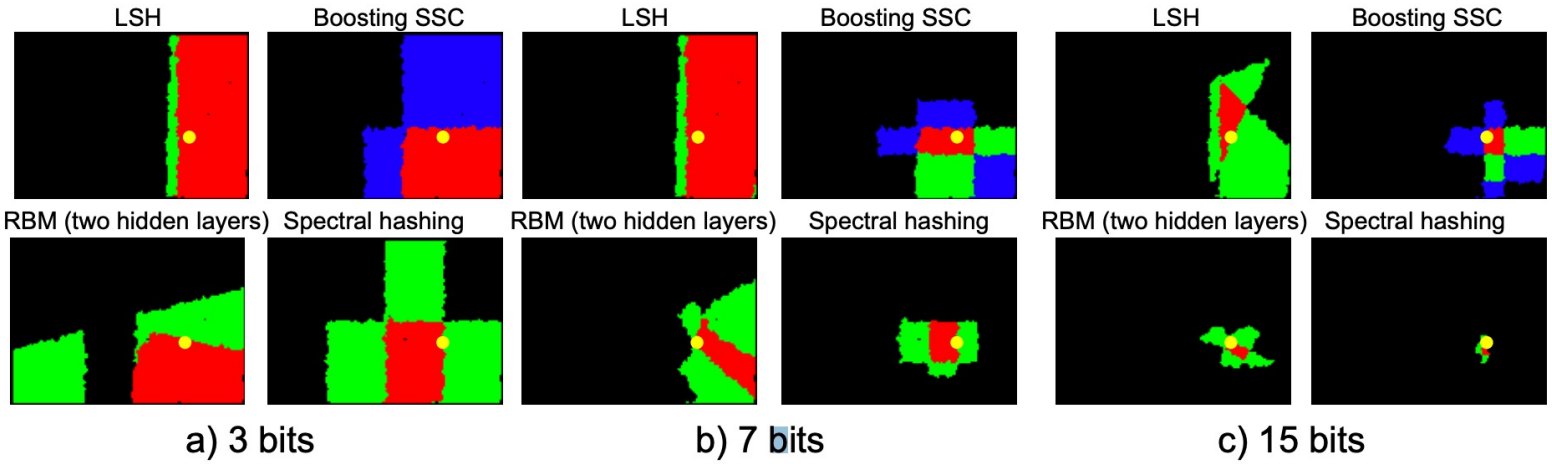
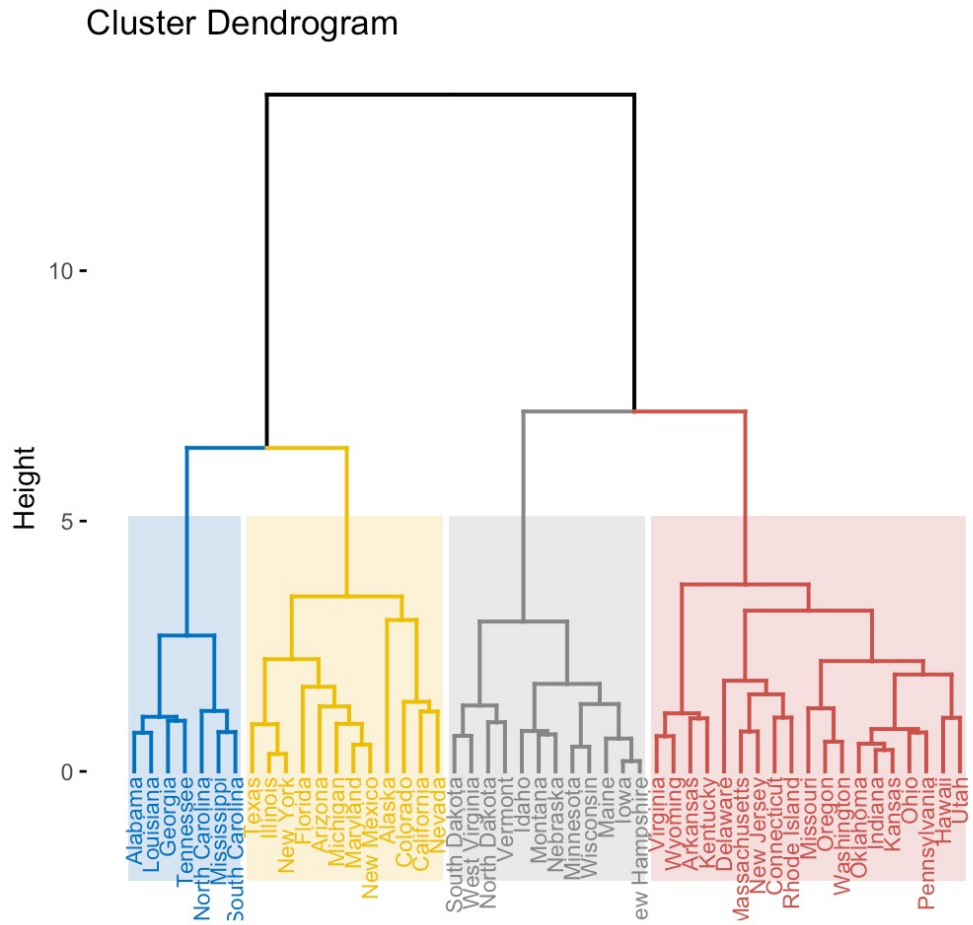
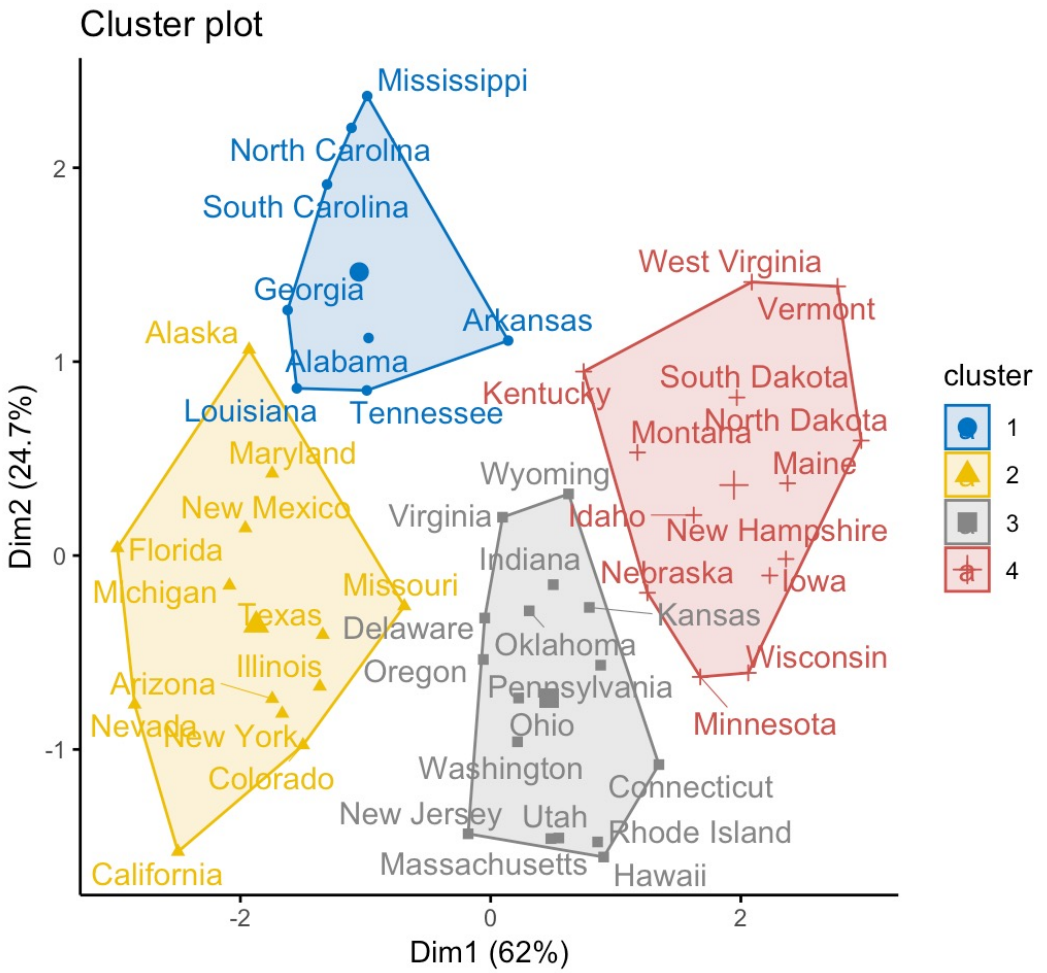


Figure 3: Comparison of neighborhood defined by hamming balls of different radii using codes obtained with LSH, Boosting, RBM and spectral hashing when using 3, 7 and 15 bits. The yellow dot denotes a test sample. The red points correspond to the locations that are within a hamming distance of zero. Green corresponds to a hamming ball of radius 1, and blue to radius 2.

Background: Hierarchical k-means

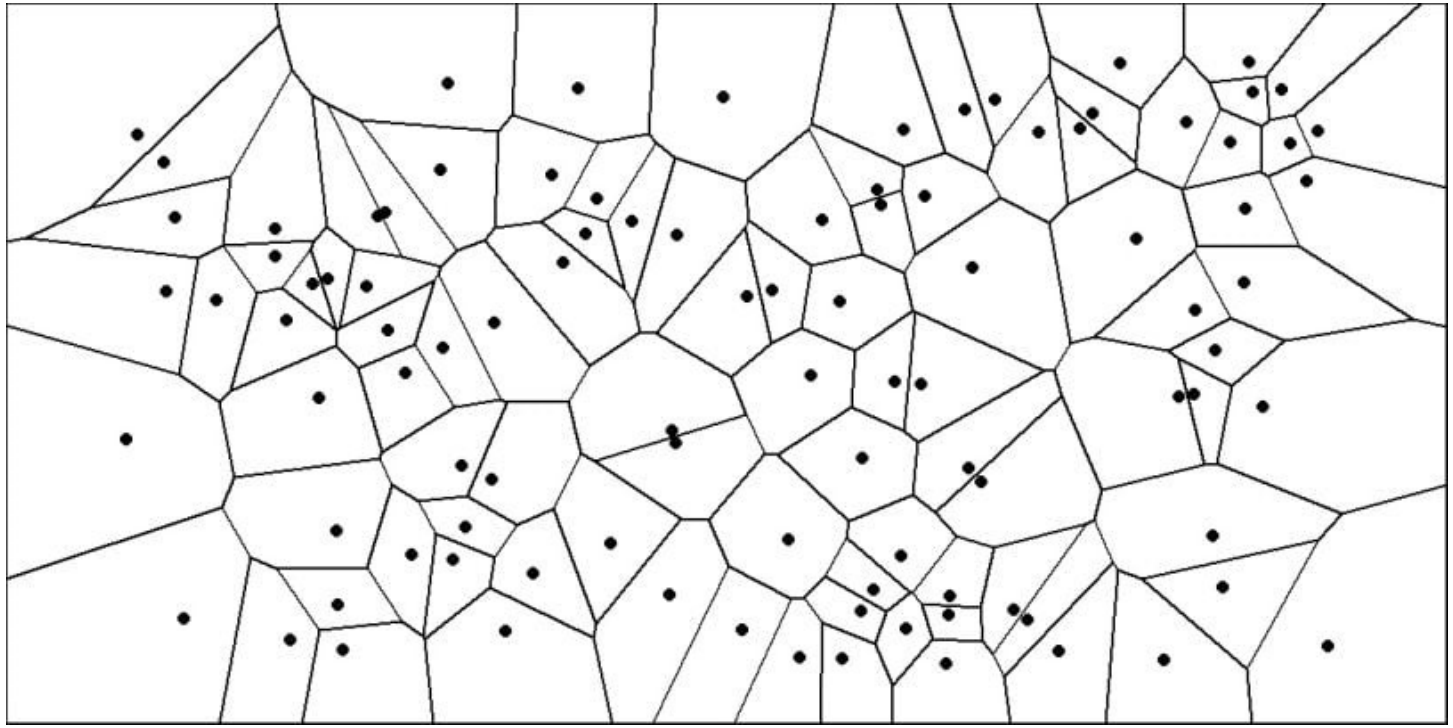




images from https://www.datanovia.com/en/lessons/hierarchical-k-means-clustering-optimize-clusters/

Product Quantization for ANN

Quantization forming Voronoi diagram



c_i = centroid

C = codebook

One algorithm:
basic k-means clustering works

Problem:
# of centroids = 2^(bits)

Product Quantization

$$\underbrace{x_1, \ldots, x_{D^*}}_{u_1(x)}, \ldots, \underbrace{x_{D-D^*+1}, \ldots, x_D}_{u_m(x)} \rightarrow q_1\big(u_1(x)\big), \ldots, q_m\big(u_m(x)\big), \tag{8}$$

$$\mathcal{C} = \mathcal{C}_1 \times \ldots \times \mathcal{C}_m,$$

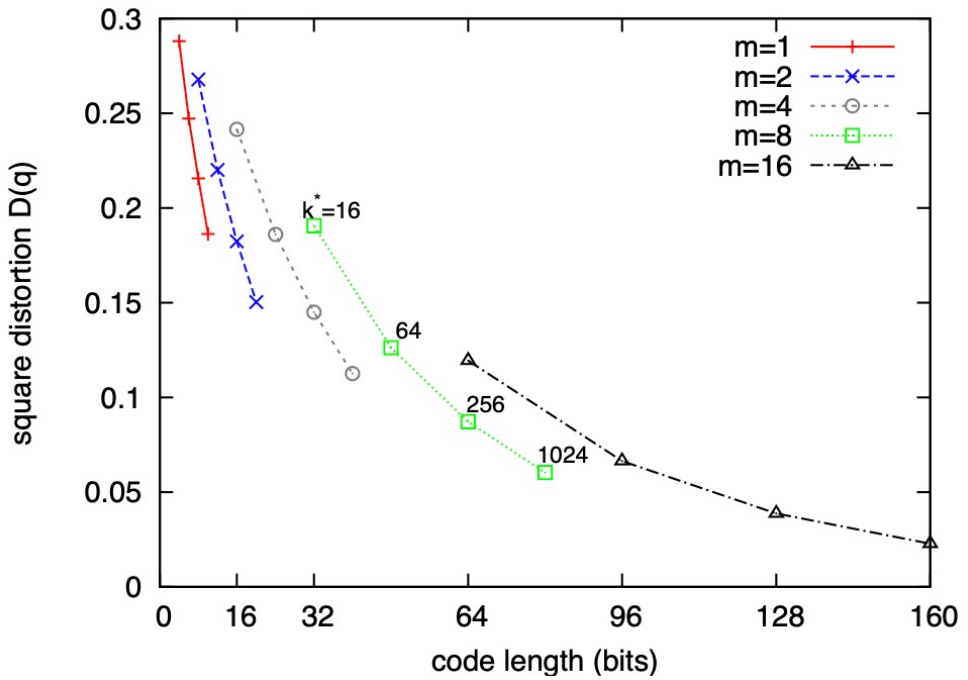
Can produce a large number of centroids from several small sets of centroids

Product Quantization Complexity

	memory usage	assignment complexity
k-means	$k\,D$	$k\,D$
HKM	$\frac{b_\mathrm{f}}{b_\mathrm{f}-1}(k-1)\,D$	$l\,D$
product k-means	$m\,k^*\,D^* = k^{1/m}\,D$	$m\,k^*\,D^* = k^{1/m}\,D$

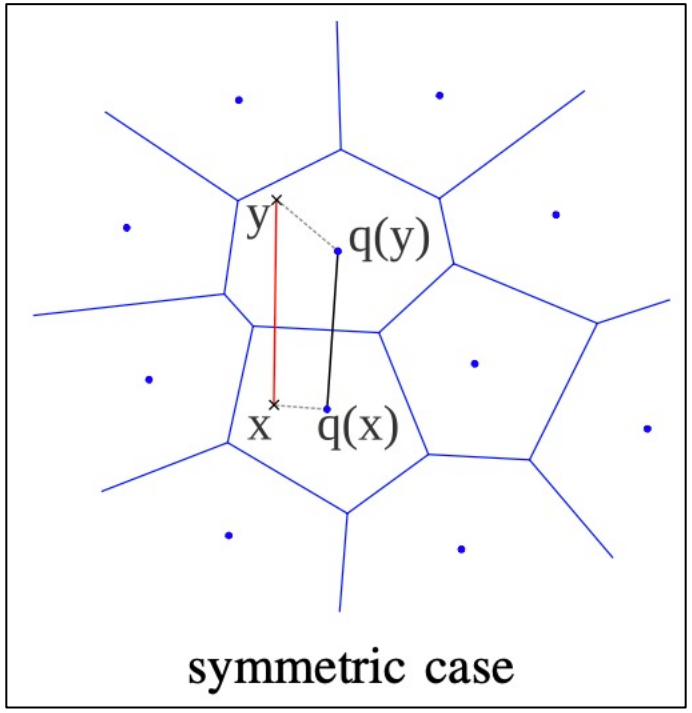
Product Quantization Error

$$\mathrm{MSE}(q) = \sum_j \mathrm{MSE}(q_j), \qquad (11)$$



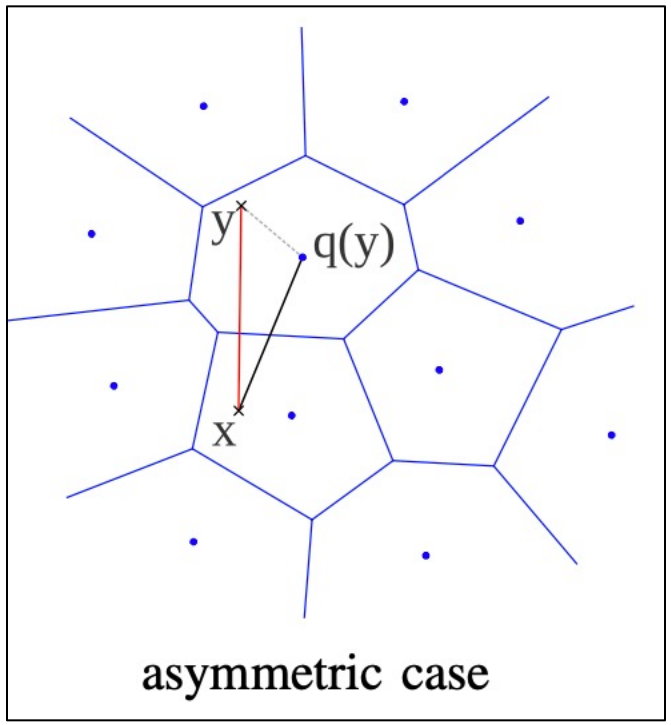
Searching: Symmetric

$$\hat{d}(x,y) = d\big(q(x), q(y)\big) = \sqrt{\sum_j d\big(q_j(x), q_j(y)\big)^2},$$

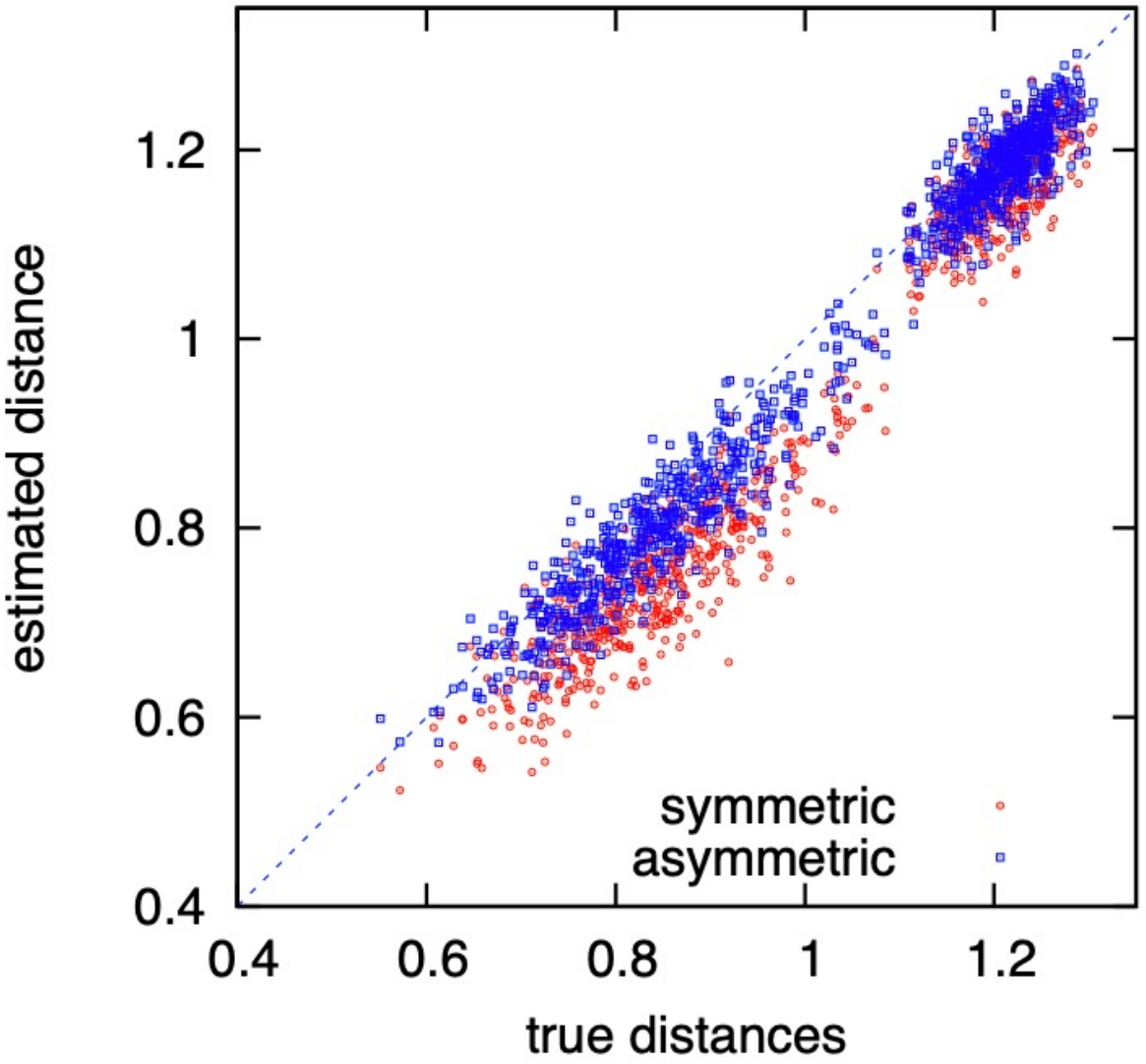


Searching: Asymmetric (preferred)

$$\tilde{d}(x, y) = d\big(x, q(y)\big) = \sqrt{\sum_j d\big(u_j(x), q_j(u_j(y))\big)^2}, \tag{13}$$

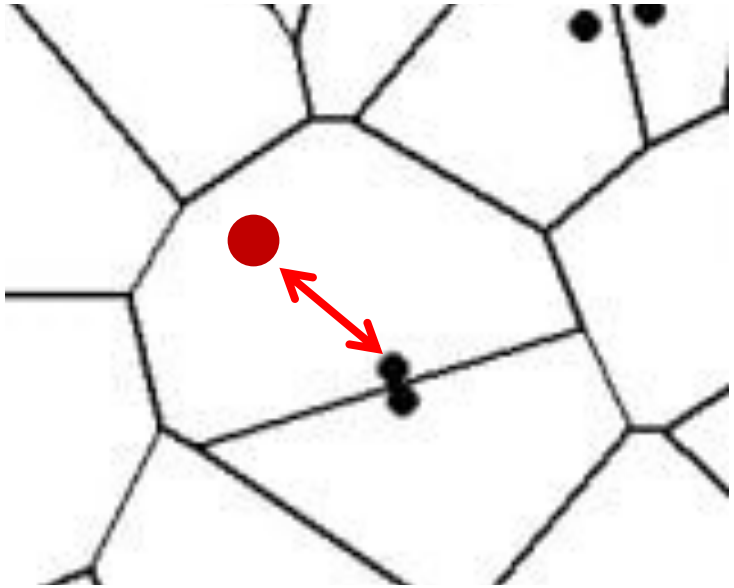


Bias: Asymmetric vs Symmetric



Search with Product Quantization (IVF-ADC)

- PQ is used for estimating *residual*



Search with Product Quantization

- PQ is used for estimating *residual*
- First, we use a *coarse quantizer* (k-means with a small *k*)

$$r(y) = y - q_c(y), \tag{28}$$

- Second, the residual is encoded using PQ

$$\ddot{y} \triangleq q_c(y) + q_p\big(y - q_c(y)\big). \tag{29}$$

x is a query, *y* is a database vector

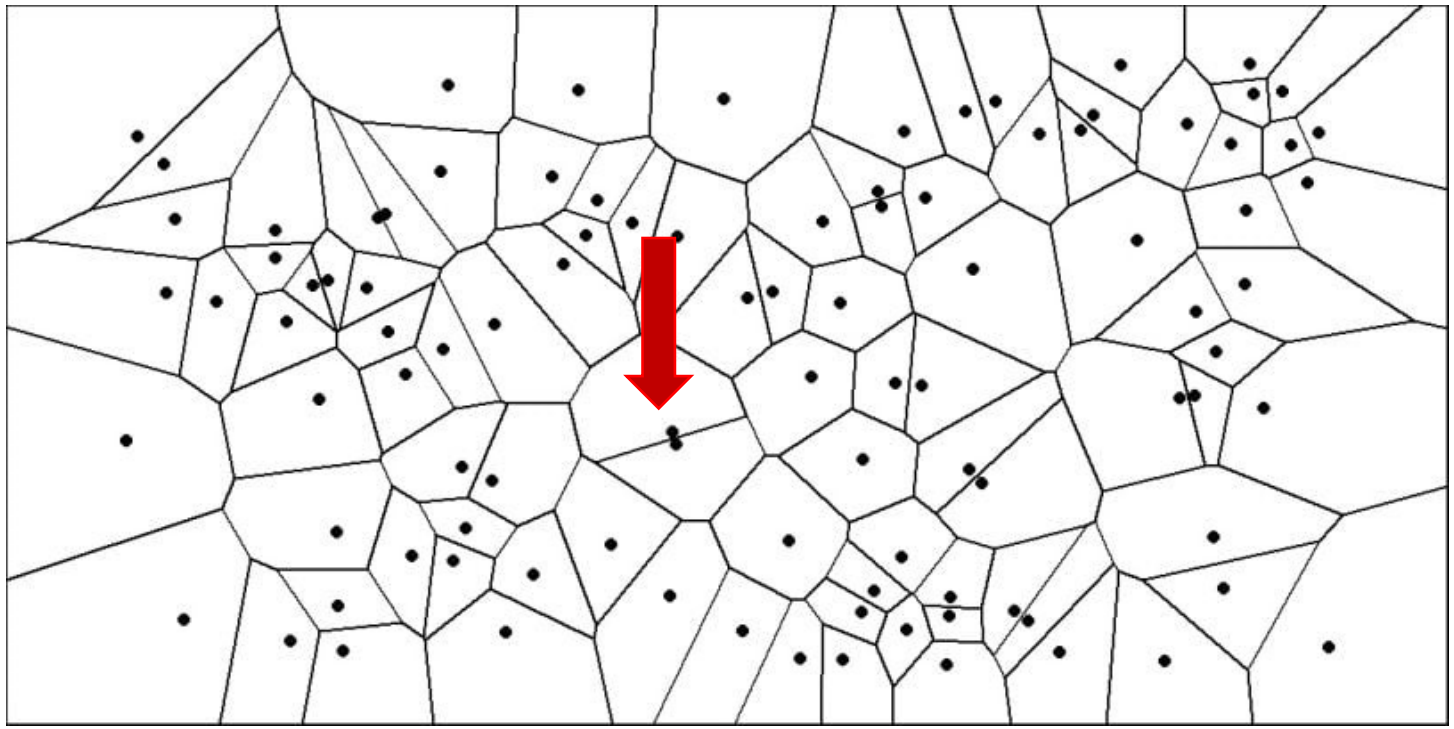
Search with Product Quantization

- Compute distance between x and $\ddot{y} \triangleq q_{\mathrm{c}}(y) + q_{\mathrm{p}}\big(y - q_{\mathrm{c}}(y)\big).$ (29)
- Compute distance between $x - q_c(y)$ and $y - q_c(y)$

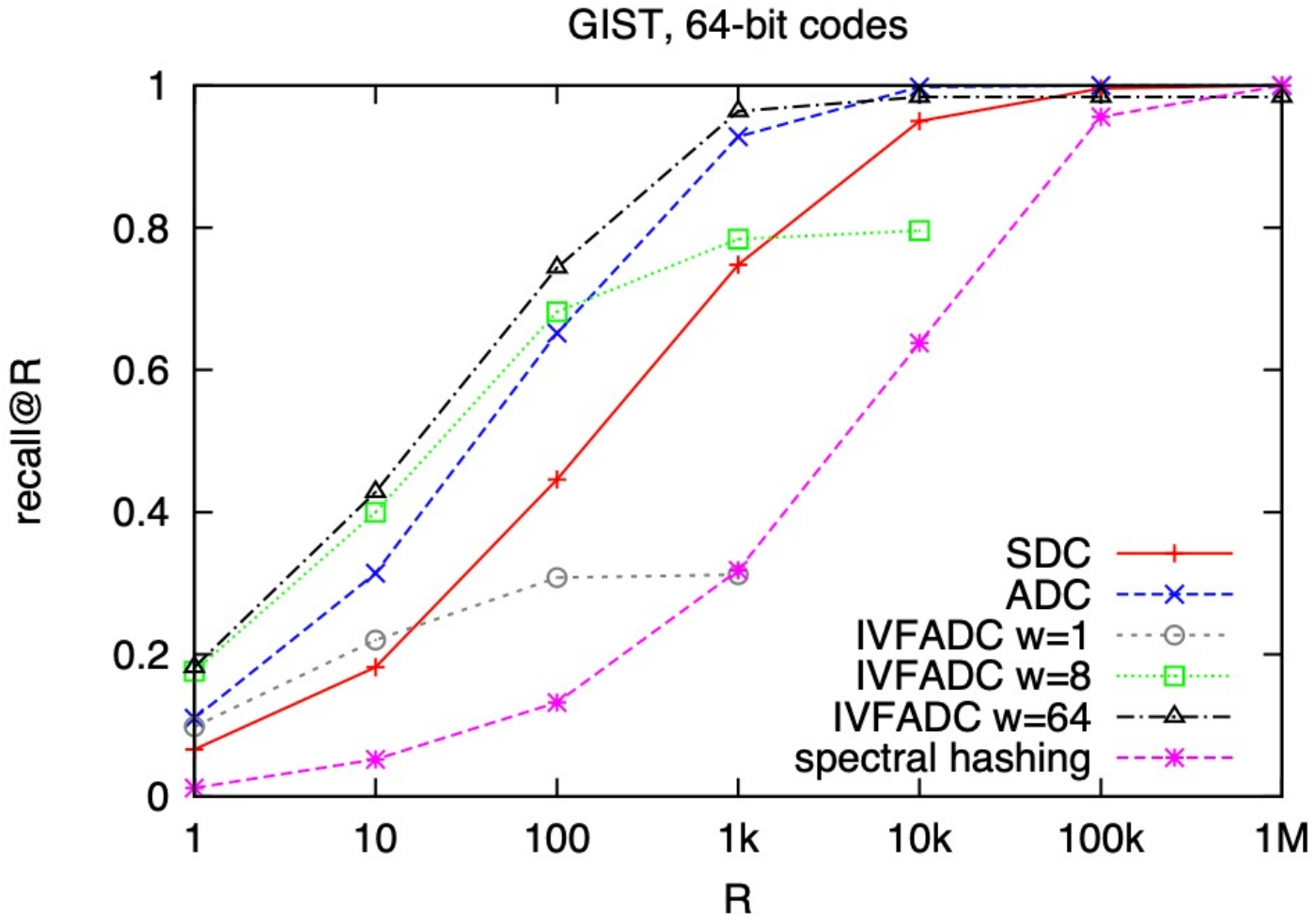
$$\ddot{d}(x, y) = d(x, \ddot{y}) = d\Big(x - q_{\mathrm{c}}(y), q_{\mathrm{p}}\big(y - q_{\mathrm{c}}(y)\big)\Big). \quad (30)$$

x is a query, *y* is a database vector

Search neighbors (how many? *w*)



Evaluation



w := # of cells to examine

Summary

- Production Quantization (PQ) is an efficient ANN method
- Tends to work better than LSH as it considers distribution
- We will learn graph-based methods: HNSW and others

Questions?