

Acknowledgments

The research project is implemented in the framework of H.F.R.I call “3rd Call for H.F.R.I.” Research Projects to Support Faculty Members & Researchers” (H.F.R.I. Project Number: 25146).



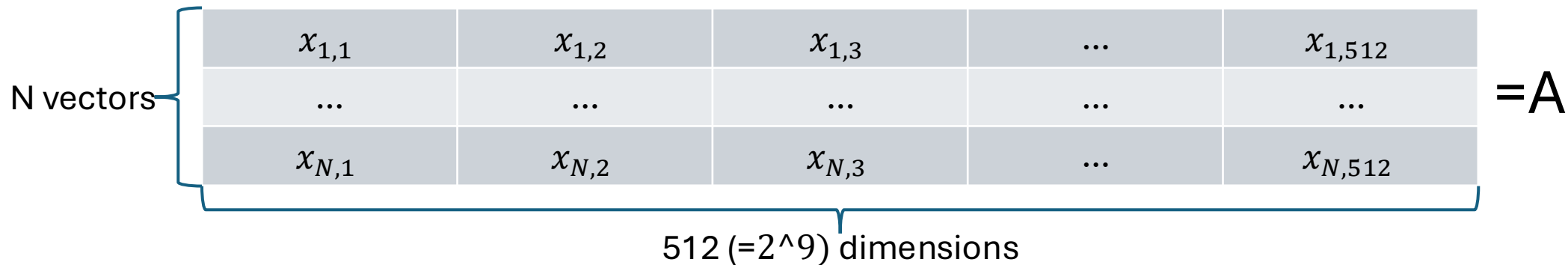
Quantization Methods for Data Compression

Dimitris Petratos



Problem Statement

- High-dimensional floating points → memory cost



If every $x_{i,j}$ value is stored as a 32-bit floating point (4-byte), then A needs

$$2^9 \cdot N \cdot 4 \text{ bytes} = 2^{11} \cdot N \text{ bytes}$$

If $N = 10^8$, then we should have $\sim 205GB$ memory storage.

Quantization Solution

- If we could represent each $x_{i,j}$ 32FP value as INT8 (8-bit integer), then the storage needed would be

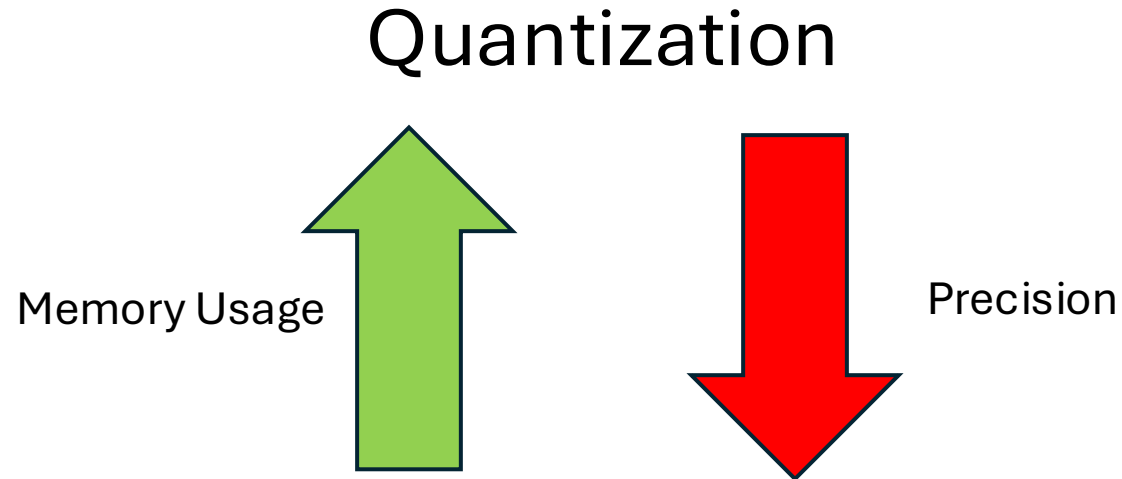
$$2^9 \cdot N \cdot 1 \text{ bytes} = 2^9 \cdot N \text{ bytes}$$

And for the same $N = 10^8$, we would need $\sim 51GB$ memory storage.

- That's what **quantization methods** try to achieve.

Obvious drawbacks

- Compressing 32FP vectors into INT8 vectors, lead to precision loss.



Quantization Methods

- Codebook-based methods:
 - Product Quantization (PQ)
 - Additive Quantization (AQ)
- Dimension-wise methods:
 - Scalar Quantization (SQ)
 - Binary Quantization (BQ)

Preliminaries

- A quantizer function $q: \mathbb{R}^d \rightarrow \mathcal{C}$, is a function that maps a vector $x \in \mathbb{R}^d$, to a set of indices $q(x) \in \mathcal{C} = \{c_i: i \in I\}$, where I is a finite set ($I = 0, 1, \dots, k - 1$).
- The reproduction values c_i are called **centroids** and the set of reproduction values $\mathcal{C}$ is called the **codebook** of size k .
- The set $V_i = \{x \in \mathbb{R}^d: q(x) = c_i\}$ is called a (Voronoi) cell and the set of cells of a quantizer function forms a partition in $\mathbb{R}^d$.

Codebook-based Methods



segments

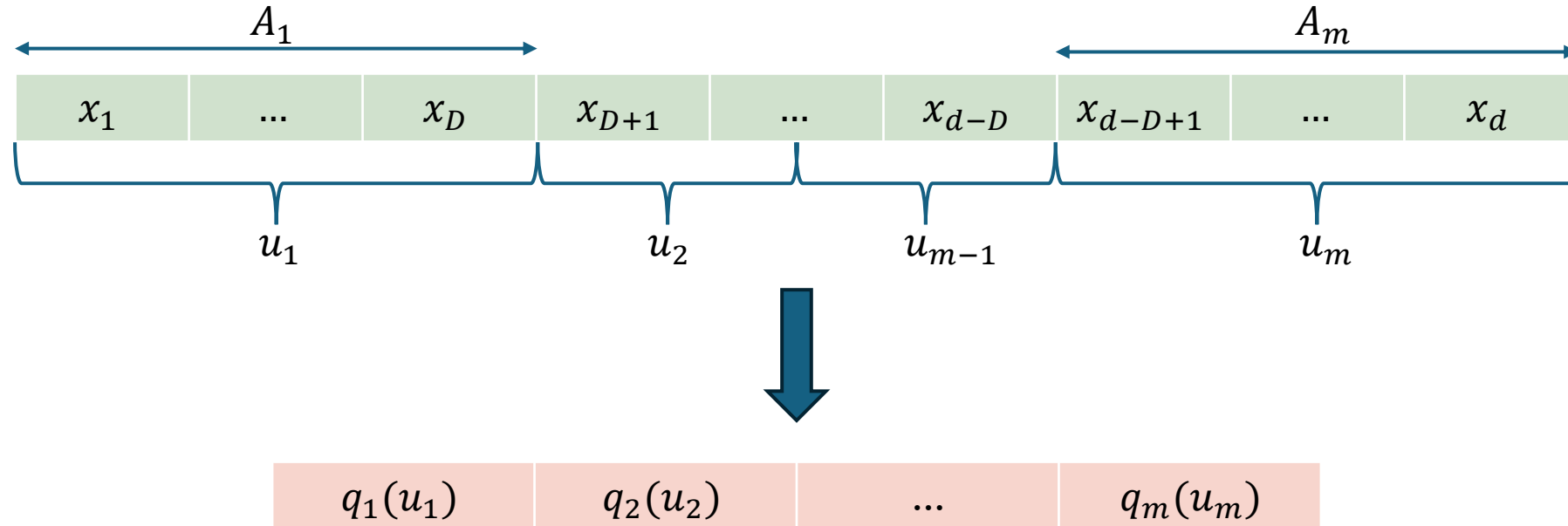
- Product Quantization
- Additive Quantization

They perform transformations per “buckets” of dimensions.

- H. Jegou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(1):117–128, 2011.
- A. Babenko and V. Lempitsky. Additive quantization for extreme vector compression. In 2014 IEEE Conference on Computer Vision and Pattern Recognition, pages 931–938, 2014.

Product Quantization - Overview

$$D = d/m$$



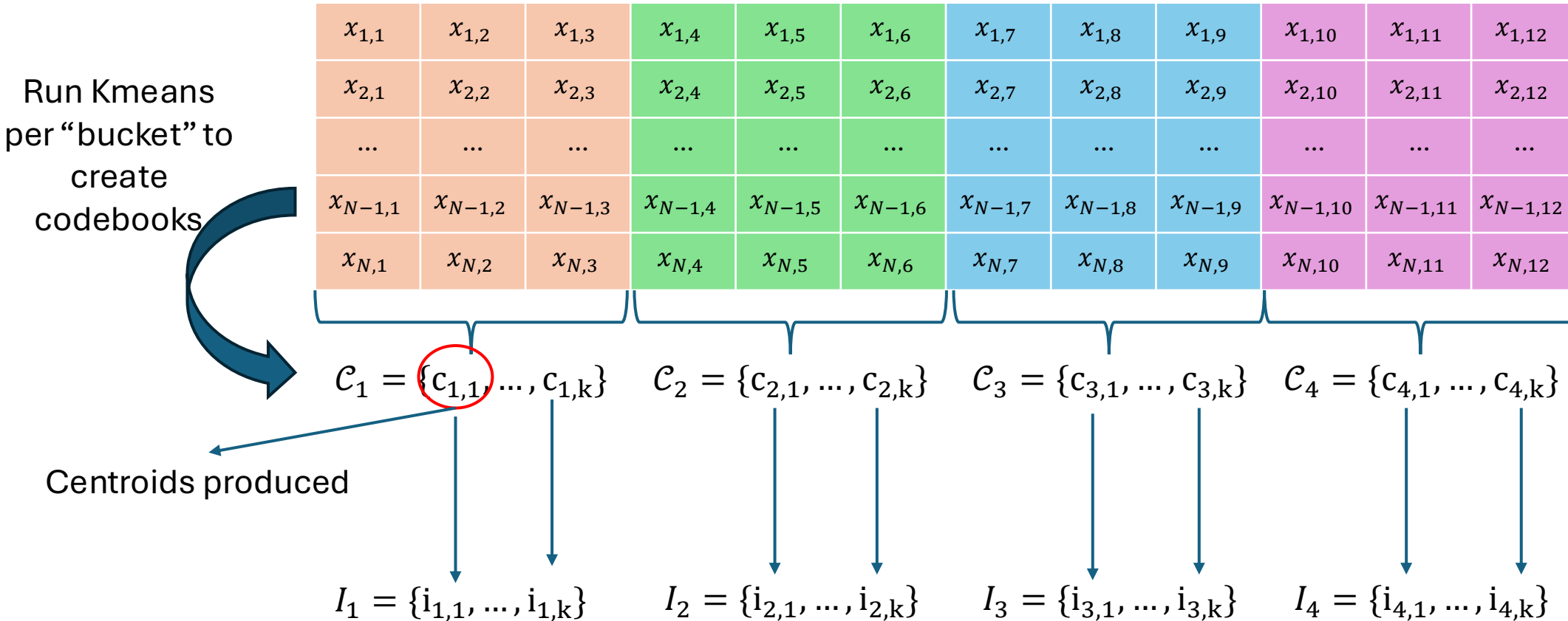
For every dimension “bucket” j , q_j is associated with:

- An index set I_j
- A codebook $\mathcal{C}_j$

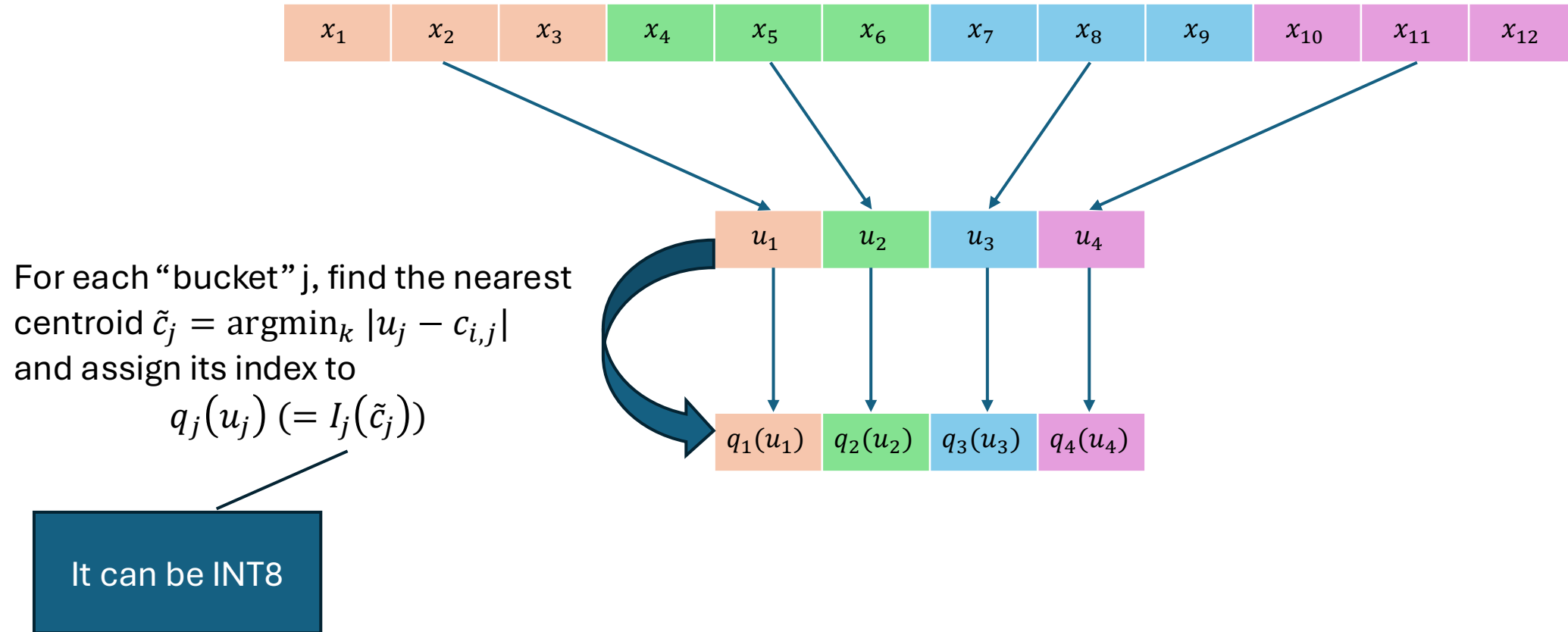
So essentially, PQ builds quantizer functions for “buckets” of dimensions whose Cartesian product constitutes the original space: $\prod_{j=1}^m A_j = \mathbb{R}^d$.

In this way it creates a codebook for $x \in \mathbb{R}^d$, defined as $\mathcal{C} = \mathcal{C}_1 \times \dots \times \mathcal{C}_m$.

Product Quantization – An example



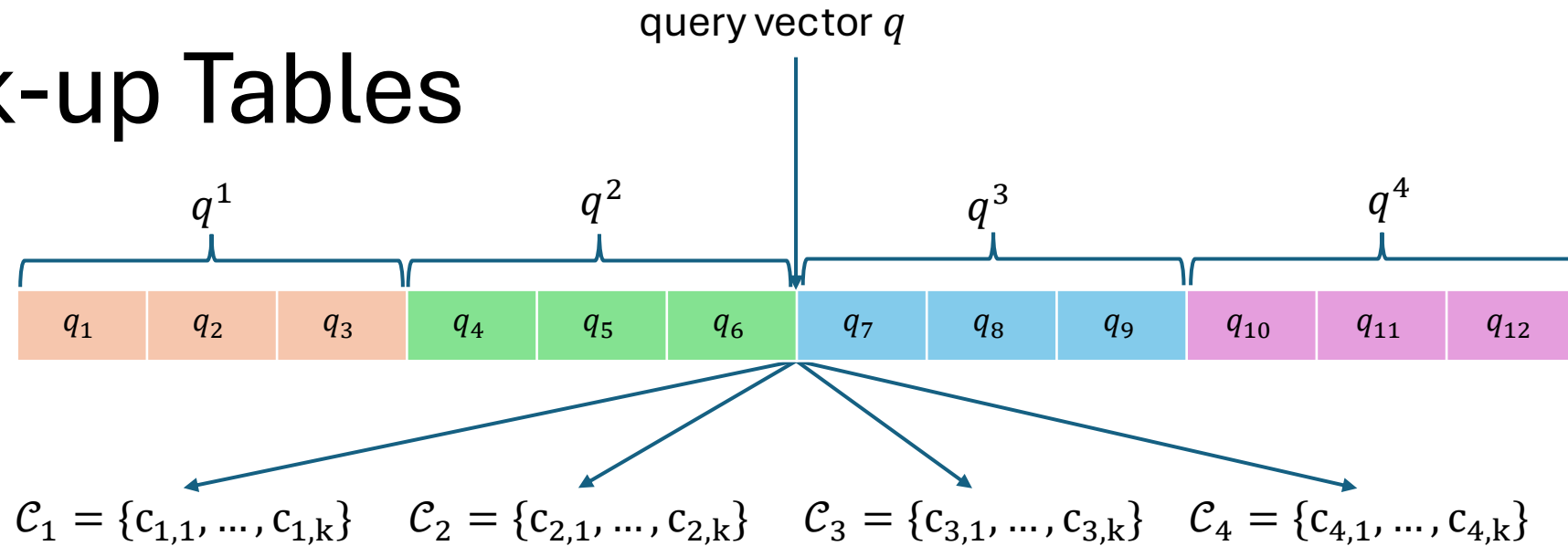
Product Quantization – Final Compression



Product Quantization – Distance Computation

- Symmetric Distance Computation (SDC):
$$\delta(x, y) = \delta(q(x), q(y))$$
- Asymmetric Distance Computation (ADC):
$$\delta(x, y) = \delta(x, q(y))$$

Look-up Tables



$d(q^1, c_{1,1})$	$d(q^2, c_{2,1})$	$d(q^3, c_{3,1})$	$d(q^4, c_{4,1})$
$d(q^1, c_{1,2})$	$d(q^2, c_{2,2})$	$d(q^3, c_{3,2})$	$d(q^4, c_{4,2})$
...	...	...	...
$d(q^1, c_{1,k})$	$d(q^2, c_{2,k})$	$d(q^3, c_{3,k})$	$d(q^4, c_{4,k})$

VDB vector: $x = (x^1, x^2, x^3, x^4)$

Additive Quantization - Overview

- Does not assume any independence between subspaces
- Codebooks ($\mathcal{C}^m = \{c^m(i_m) : i_m \in I_m, |I_m| = K\}, m = 1, \dots, M$) are generated through an optimization problem
- Vector transformation: $x \approx \sum_{m=1}^M c^m(i_m)$

$$\min_{\substack{C^1, \dots, C^M \subset \mathbf{R}^D \\ |C^m| = K \\ i_j^m \in 1..K}} \sum_{j=1}^n \left\| x_j - \sum_{m=1}^M c^m(i_j^m) \right\|^2$$

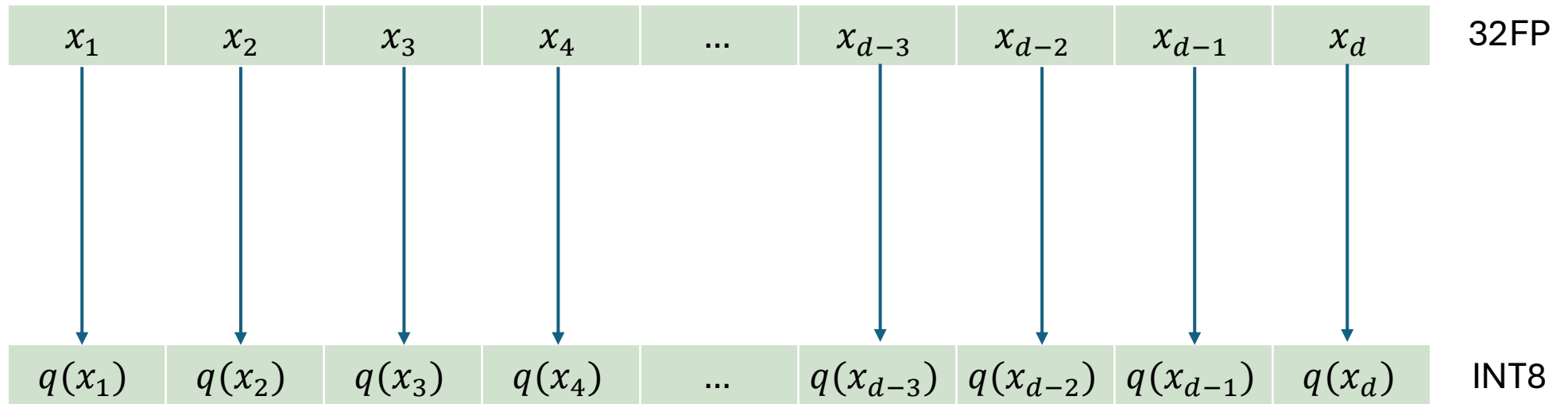
Dimension-wise Methods

- Scalar Quantization
- Binary Quantization

They perform transformations per dimension and not for “buckets” of dimensions.

- J. Wang, X. Yi, R. Guo, H. Jin, P. Xu, S. Li, X. Wang, X. Guo, C. Li, X. Xu, et al. Milvus: A purpose-built vector data management system. In Proceedings of the 2021 International Conference on Management of Data, pages 2614–2627, 2021.
- W. Zhou, Y. Lu, H. Li, and Q. Tian. Scalar quantization for large scale image search. In Proceedings of the 20th ACM International Conference on Multimedia, MM '12, page 169–178, New York, NY, USA, 2012. Association for Computing Machinery.

Scalar Quantization - Overview



$$\text{SQ8: } q(x_i) = \left\lfloor 255 \cdot \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \right\rfloor$$

Scalar Quantization - Example

x	7.37	-2.62	4.05	6.11	6.91
y	-4.92	-3.93	5.71	2.05	-2.83
z	6.12	8.27	6.9	-0.52	1.03
h	-0.87	-0.25	-2.77	4.04	0.60

SQ8

$q(x)$	255	27	179	255	255
$q(y)$	0	0	223	98	0
$q(z)$	229	255	255	0	101
$q(h)$	84	76	0	175	89

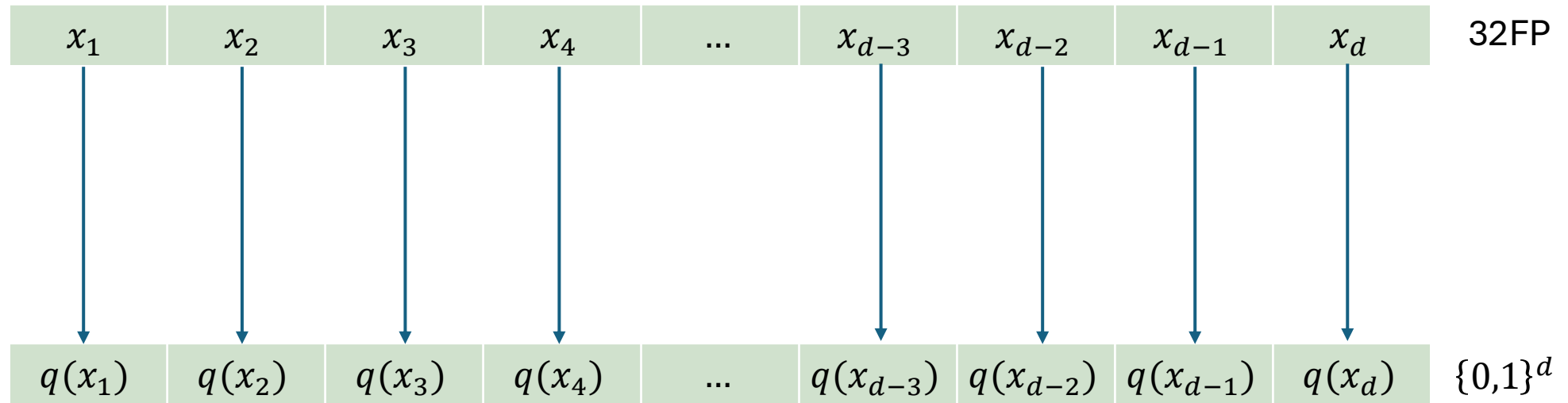
Vectors	1-NN	2-NN	3-NN
x	h	z	y
y	h	x	z
z	x	h	y
h	y	x	z

Euclidean distance

Vectors	1-NN	2-NN	3-NN
$q(x)$	$q(h)$	$q(z)$	$q(y)$
$q(y)$	$q(h)$	$q(z)$	$q(x)$
$q(z)$	$q(y)$	$q(x)$	$q(h)$
$q(h)$	$q(y)$	$q(x)$	$q(z)$

SDC

Binary Quantization - Overview



$$q(x_i) = \begin{cases} 0, & x_i \leq \tau_i \\ 1, & x_i > \tau_i \end{cases}$$

Binary Quantization - Example

x	7.37	-2.62	4.05	6.11	6.91
y	-4.92	-3.93	5.71	2.05	-2.83
z	6.12	8.27	6.9	-0.52	1.03
h	-0.87	-0.25	-2.77	4.04	0.60

BQ ($\tau_i = 2$)

$q(x)$	1	0	1	1	1
$q(y)$	0	0	1	1	0
$q(z)$	1	1	1	0	0
$q(h)$	0	0	0	1	0

Vectors	1-NN	2-NN	3-NN
x	h	z	y
y	h	x	z
z	x	h	y
h	y	x	z

Euclidean distance

Vectors	1-NN	2-NN	3-NN
$q(x)$	$q(y)$	$q(z)$	$q(h)$
$q(y)$	$q(h)$	$q(x)$	$q(z)$
$q(z)$	$q(x)$	$q(y)$	$q(h)$
$q(h)$	$q(y)$	$q(x)$	$q(z)$

SDC

Acknowledgments

The research project is implemented in the framework of H.F.R.I call “3rd Call for H.F.R.I.” Research Projects to Support Faculty Members & Researchers” (H.F.R.I. Project Number: 25146).

