Advances in Product Quantization

RaBitQ & TurboQuant as examples

大数据算法课程助教
陈磊

中国科学技术大学
University of Science and Technology of China

Contents

- Discussions
- Introduction for PQ
- Background and notations
- RabitQ
- TurboQuant

01 Discussions

Discussions on OpenReview

- Proceedings of the ACM on Management of Data, 2024

RaBitQ: Quantizing High-Dimensional Vectors with a Theoretical Error Bound for Approximate Nearest Neighbor Search

Jianyang Gao
jianyang.gao@ntu.edu.sg
Nanyang Technological University
Singapore

Cheng Long*
c.long@ntu.edu.sg
Nanyang Technological University
Singapore

- ICLR 2026 Poster

TurboQuant: Online Vector Quantization with Near-optimal Distortion Rate

Amir Zandieh
Google Research
zandieh@google.com

Majid Daliri
New York University
daliri.majid@nyu.edu

Majid Hadian
Google DeepMind
majidh@google.com

Vahab Mirrokni
Google Research
mirrokni@google.com

Discussions on OpenReview

- Comparisons between different strategies for (1) codebook construction and (2) unbiased estimations.
- Imcomplete discussion for RabitQ?
- Ungrounded theoretical discussion?
- Transparent empirical comparisons required?

- Dive into these two papers, learn some new algorithm designs, and something to take care about.

TurboQuant https://openreview.net/forum?id=tO3ASKZlok

Introduction for Product Quantization (PQ)

- Why product? Dimension reduction and clustering.
- Mutual transformation between 2-norms and cosine similarity:

$$\|\mathbf{o}_r - \mathbf{q}_r\|^2 = \|(\mathbf{o}_r - \mathbf{c}) - (\mathbf{q}_r - \mathbf{c})\|^2$$

$$= \|\mathbf{o}_r - \mathbf{c}\|^2 + \|\mathbf{q}_r - \mathbf{c}\|^2 - 2 \cdot \|\mathbf{o}_r - \mathbf{c}\| \cdot \|\mathbf{q}_r - \mathbf{c}\| \cdot \langle \mathbf{q}, \mathbf{o} \rangle$$

- Application:
  - attention scores, recommendations.
  - Kv cache quantization, information retrieval (RAG)
  - Approximation nearest neighbor search (ANN)

Introduction for PQ

- Product quantization (PQ) is a family of popular methods for approximate nearest neighbors (ANN), while ANN is a relaxed counterpart of NN.
- The quality of ANN (PQ algorithms) affects the recall rate after re-ranking (in recommendation system).
- Given data vectors and query vector(s).
- Two phases in PQ:
  - Index phase: construct quantization codebook, map each data vector to its quantized vector.
  - Query phase: pre-compute <query vector, quantized vectors>, assign as the estimated distance of each data vectors.

Challenges of PQ and proposal of RabitQ

- The same estimated distance for data vectors within same cluster. (Does RabitQ changes that?)
- Hard to quantify the approximation error during codebook construction and distance estimation
- It implies that PQ can unpredictably fail anytime, moderately or severely.
- RabitQ:
  - Provide geometric relationship between <query, data> and <query, quantified data>.
  - Provide unbiased estimator for <query, data> based on that geometric relationship.
  - Provide error bound analysis.
  - Practical implementations.

Basic idea of RabitQ

- Normalize: the normalized data vectors are expected to spread evenly on the unit hypersphere
- Intuitively construct codebook: also spread evenly on the unit hypersphere (and think about the benefit in theory)
- Inject some random rotations to remove inductive bias.
- Assign Quantized Codes of Data Vectors.
    - Index phase ↑ | query phase ↓
- Quantizing query vector (uniform scalar quantization with loglogD resolution and randomness)
- Efficiently compute as a weighted sum of the inner product of the binary vectors

RabitQ preliminary

Normalize	$\mathbf{o} := \frac{\mathbf{o}_r - \mathbf{c}}{\|\mathbf{o}_r - \mathbf{c}\|}$	$\mathbf{o}_r, \mathbf{q}_r$	The raw data and query vectors.
		$\mathbf{o}, \mathbf{q}$	The normalized data and query vectors.
Construct	$C := \left\{+\frac{1}{\sqrt{D}}, -\frac{1}{\sqrt{D}}\right\}^D$	C, C_{rand}	The quantization codebook, its randomized version.
		P	A random orthogonal transformation matrix.
	$C_{rand} := \{P\mathbf{x} \mid \mathbf{x} \in C\}$	$\bar{\mathbf{x}}$	The code in C s.t. $P\bar{\mathbf{x}}$ is the quantized vector of $\mathbf{o}$.
Assign by Picking	$\bar{\mathbf{x}} = \arg\min_{\mathbf{x} \in C} \|\mathbf{o} - P\mathbf{x}\|^2$	$\bar{\mathbf{o}}$	The quantized vector of $\mathbf{o}$ in C_{rand}, i.e., $\bar{\mathbf{o}} = P\bar{\mathbf{x}}$.
		$\bar{\mathbf{x}}_b$	The quantization code of $\mathbf{o}$ as a D-bit string.
	Sign($\bar{x}$)=Sign(P^{-1}o)	$\mathbf{q}'$	The inversely transformed query vector, i.e., $P^{-1}\mathbf{q}$.
		$\bar{\mathbf{q}}$	The quantized query vector of $\mathbf{q}'$.
Compute	$\langle \bar{\mathbf{x}}_b, \bar{\mathbf{q}}_u \rangle$	$\bar{\mathbf{q}}_u$	The unsigned integer representation of $\bar{\mathbf{q}}$.

Approximation error of RabitQ

$$\mathbb{E}\left[\frac{\langle \bar{\mathbf{o}}, \mathbf{q}\rangle}{\langle \bar{\mathbf{o}}, \mathbf{o}\rangle}\right] = \langle \mathbf{o}, \mathbf{q}\rangle$$

$$\frac{\langle \bar{\mathbf{o}}, \mathbf{q}\rangle}{\langle \bar{\mathbf{o}}, \mathbf{o}\rangle} = \langle \mathbf{o}, \mathbf{q}\rangle + \sqrt{1 - \langle \mathbf{o}, \mathbf{q}\rangle^2} \cdot \frac{\langle \bar{\mathbf{o}}, \mathbf{e}_1\rangle}{\langle \bar{\mathbf{o}}, \mathbf{o}\rangle}$$

$$\left|\frac{\langle \bar{\mathbf{o}}, \mathbf{q}\rangle}{\langle \bar{\mathbf{o}}, \mathbf{o}\rangle} - \langle \mathbf{o}, \mathbf{q}\rangle\right| = O\left(\frac{1}{\sqrt{D}}\right) \textit{ with high probability}$$

$$\mathbb{P}\left\{\left|\frac{\langle \bar{\mathbf{o}}, \mathbf{q}\rangle}{\langle \bar{\mathbf{o}}, \mathbf{o}\rangle} - \langle \mathbf{o}, \mathbf{q}\rangle\right| > \sqrt{\frac{1 - \langle \bar{\mathbf{o}}, \mathbf{o}\rangle^2}{\langle \bar{\mathbf{o}}, \mathbf{o}\rangle^2}} \cdot \frac{\epsilon_0}{\sqrt{D-1}}\right\} \leq 2e^{-c_0\epsilon_0^2}$$

Approximation error of RabitQ

$$\langle \bar{\mathbf{o}}, \mathbf{q} \rangle = \langle P\bar{\mathbf{x}}, \mathbf{q} \rangle = \left\langle P^{-1}P\bar{\mathbf{x}}, P^{-1}\mathbf{q} \right\rangle = \langle \bar{\mathbf{x}}, \mathbf{q}' \rangle$$

THEOREM 3.3. *$B_q = \Theta(\log\log D)$ suffices to guarantee that $|\langle \bar{\mathbf{x}}, \mathbf{q}' \rangle - \langle \bar{\mathbf{x}}, \bar{\mathbf{q}} \rangle| = O(1/\sqrt{D})$ with high probability.*

Experiments of RabitQ

Time in the Indexing Phase

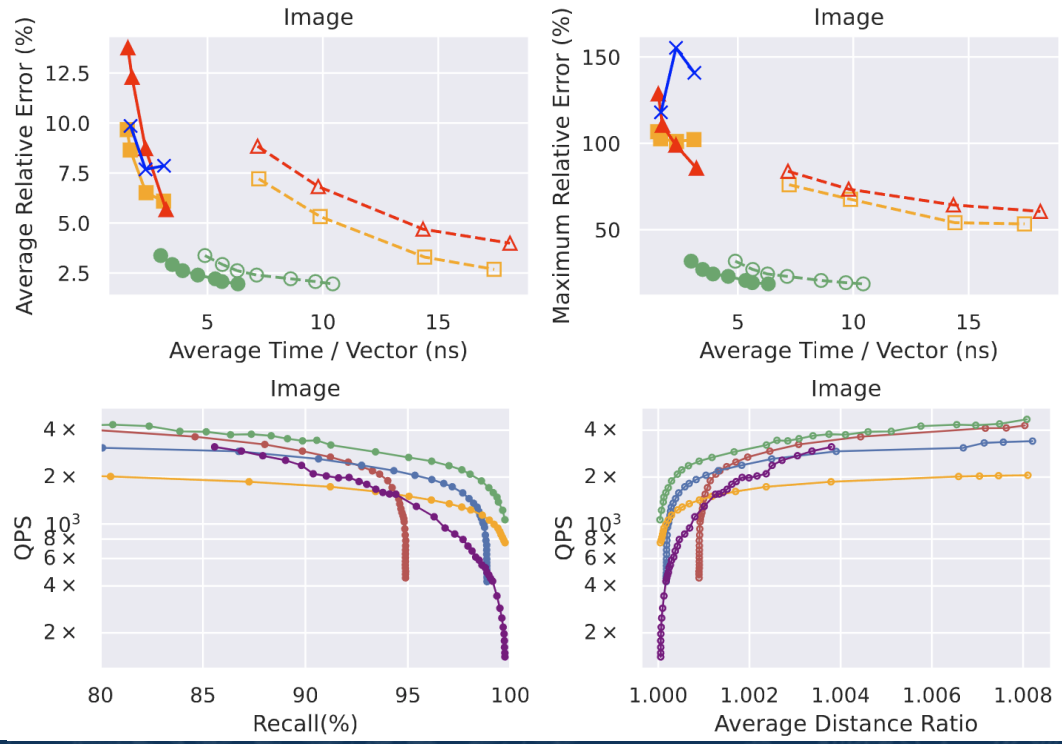
Table 4: The Indexing Time for the GIST Dataset

	RaBitQ	PQ	OPQ	LSQ
Time	117s	105s	291s	time-out (>24 hours)

Time-Accuracy Trade-Off per Vector for

Distance Estimation (first row)

ANN Search (second row)



Experiments of RabitQ

"the theoretical analysis provides explicit suggestions on the parameters. Thus, our method requires no tuning."
However…

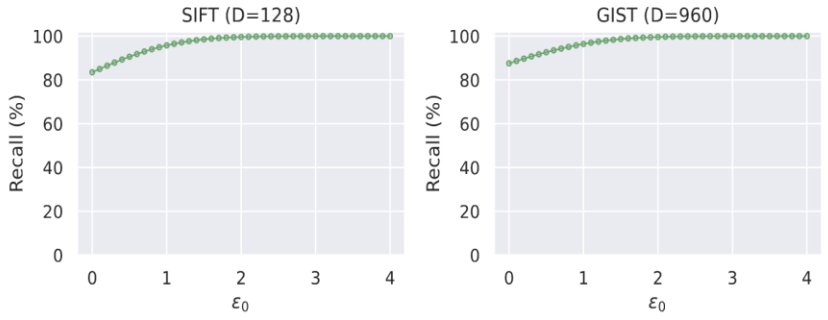


Figure 5: Verification Study on ϵ_0.

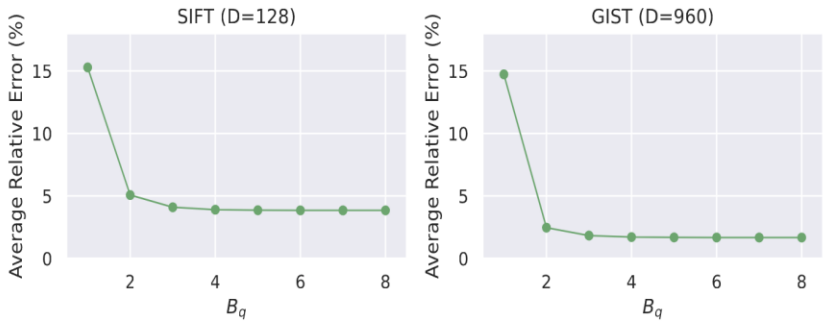


Figure 6: Verification Study on B_q.

Because, they empirically use a confidence interval to decide which samples to re-rank or drop out in practice.

$$\frac{\langle \bar{\mathbf{o}}, \mathbf{q} \rangle}{\langle \bar{\mathbf{o}}, \mathbf{o} \rangle} \pm \sqrt{\frac{1 - \langle \bar{\mathbf{o}}, \mathbf{o} \rangle^2}{\langle \bar{\mathbf{o}}, \mathbf{o} \rangle^2}} \cdot \frac{\epsilon_0}{\sqrt{D-1}}$$

TurboQuant

- Define **different** objectives for cosine similarity and 2-norm, quantizers optimized for MSE do not produce unbiased estimators for inner products in their cases due to its **definitions.**

$$\textbf{(MSE)} \qquad D_{\texttt{mse}} := \mathop{\mathbb{E}}_{Q}\left[\left\|\boldsymbol{x} - Q^{-1}\left(Q(\boldsymbol{x})\right)\right\|_2^2\right]$$

$$\textbf{(inner-prod error)} \qquad D_{\texttt{prod}} := \mathop{\mathbb{E}}_{Q}\left[\left|\langle \boldsymbol{y}, \boldsymbol{x}\rangle - \langle \boldsymbol{y}, Q^{-1}\left(Q(\boldsymbol{x})\right)\rangle\right|^2\right].$$

- Reweighting -> quantize the residual term

providing the following unbiased inner product estimator:

$$\left\langle \boldsymbol{y}, Q_{\texttt{mse}}^{-1}\left(Q_{\texttt{mse}}(\boldsymbol{x})\right)\right\rangle + \|\boldsymbol{r}\|_2 \cdot \left\langle \boldsymbol{y}, Q_{\texttt{qjl}}^{-1}\left(Q_{\texttt{qjl}}(\boldsymbol{r})\right)\right\rangle.$$

More formally, the quantization map $Q_{\texttt{prod}} : \mathbb{S}^{d-1} \to [2^{b-1}]^d \times \{-1, 1\}^d \times \mathbb{R}$ is defined as:

$$Q_{\texttt{prod}}(\boldsymbol{x}) = \left[Q_{\texttt{mse}}(\boldsymbol{x}), Q_{\texttt{qjl}}\left(\boldsymbol{x} - Q_{\texttt{mse}}^{-1}\left(Q_{\texttt{mse}}(\boldsymbol{x})\right)\right), \left\|\boldsymbol{x} - Q_{\texttt{mse}}^{-1}\left(Q_{\texttt{mse}}(\boldsymbol{x})\right)\right\|_2\right].$$

TurboQuant (MSE)

Algorithm 1 TurboQuant$_{\texttt{mse}}$: optimized for MSE

input: dimension d and bit-width b

// Global Parameters for Setting up TurboQuant$_{\texttt{mse}}$

Generate a random rotation matrix $\boldsymbol{\Pi} \in \mathbb{R}^{d \times d}$

Construct codebook by finding centroids $c_1, c_2, \ldots c_{2^b} \in [-1, 1]$ that minimize MSE cost in Eq. (4)

Procedure Quant$_{\texttt{mse}}(\boldsymbol{x})$

$\boldsymbol{y} \leftarrow \boldsymbol{\Pi} \cdot \boldsymbol{x}$

$\texttt{idx}_j \leftarrow \arg\min_{k \in [2^b]} |\boldsymbol{y}_j - c_k|$ for every $j \in [d]$ {$\texttt{idx}_j$'s are b-bit integers}

output: idx

Procedure DeQuant$_{\texttt{mse}}(\texttt{idx})$

$\tilde{\boldsymbol{y}}_j \leftarrow c_{\texttt{idx}_j}$ for every $j \in [d]$

$\tilde{\boldsymbol{x}} \leftarrow \boldsymbol{\Pi}^\top \cdot \tilde{\boldsymbol{y}}$

output: $\tilde{\boldsymbol{x}}$

TurboQuant (Inner Product)

Algorithm 2 TurboQuant$_{\texttt{prod}}$: optimized for inner product

input: dimension d and bit-width b

// Global Parameters for Setting up TurboQuant$_{\texttt{prod}}$

Instantiate a TurboQuant$_{\texttt{mse}}$ with bit-width $b-1$ as per Algorithm 1

Generate a random projection matrix $\boldsymbol{S} \in \mathbb{R}^{d \times d}$ with i.i.d. entries $\boldsymbol{S}_{i,j} \sim \mathcal{N}(0,1)$

Procedure Quant$_{\texttt{prod}}(\boldsymbol{x})$

$\texttt{idx} \leftarrow \text{Quant}_{\texttt{mse}}(\boldsymbol{x})$

$\boldsymbol{r} \leftarrow \boldsymbol{x} - \text{DeQuant}_{\texttt{mse}}(\texttt{idx})$ {residual vector}

$\texttt{qjl} \leftarrow \texttt{sign}(\boldsymbol{S} \cdot \boldsymbol{r})$ {QJL on residual vector}

output: $(\texttt{idx}, \texttt{qjl}, \|\boldsymbol{r}\|_2)$

Procedure DeQuant$_{\texttt{prod}}(\texttt{idx}, \texttt{qjl}, \gamma)$

$\tilde{\boldsymbol{x}}_{\texttt{mse}} \leftarrow \text{DeQuant}_{\texttt{mse}}(\texttt{idx})$

$\tilde{\boldsymbol{x}}_{\texttt{qjl}} \leftarrow \frac{\sqrt{\pi/2}}{d} \cdot \gamma \cdot \boldsymbol{S}^\top \cdot \texttt{qjl}$

output: $\tilde{\boldsymbol{x}}_{\texttt{mse}} + \tilde{\boldsymbol{x}}_{\texttt{qjl}}$

TurboQuant

- Provide error bounds for different bit-width budgets of quantizations

In Theorem 1 we prove that the b-bit MSE optimized TurboQuant $Q_{\texttt{mse}} : \mathbb{R}^d \to \{0,1\}^{b\cdot d}$ achieves the following distortion for any worst-case vector $\boldsymbol{x} \in \mathbb{R}^d$ with $\|\boldsymbol{x}\| = 1$:

- $D_{\texttt{mse}}(Q_{\texttt{mse}}) := \mathbb{E}\left[\left\|\boldsymbol{x} - Q_{\texttt{mse}}^{-1}\left(Q_{\texttt{mse}}(\boldsymbol{x})\right)\right\|_2^2\right] \leq \frac{\sqrt{3}\pi}{2} \cdot \frac{1}{4^b}$ for any $b \geq 0$.
- For small bit-widths the above distortion upper bound can be further refined. Specifically, for $b = 1, 2, 3, 4$ we have $D_{\texttt{mse}}(Q_{\texttt{mse}}) \approx \mathbf{0.36}, \mathbf{0.117}, \mathbf{0.03}, \mathbf{0.009}$, respectively.

In Theorem 2 we prove that the b-bit inner product optimized TurboQuant $Q_{\texttt{prod}} : \mathbb{R}^d \to \{0,1\}^{b\cdot d}$ achieves the following distortion for any worst-case vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ with $\|\boldsymbol{x}\| = 1$:

- $\mathbb{E}\left[\left\langle \boldsymbol{y}, Q_{\texttt{prod}}^{-1}\left(Q_{\texttt{prod}}(\boldsymbol{x})\right)\right\rangle\right] = \langle \boldsymbol{y}, \boldsymbol{x}\rangle$
- $D_{\texttt{prod}}(Q_{\texttt{prod}}) := \mathbb{E}\left[\left|\langle \boldsymbol{y}, \boldsymbol{x}\rangle - \langle \boldsymbol{y}, Q_{\texttt{prod}}^{-1}\left(Q_{\texttt{prod}}(\boldsymbol{x})\right)\rangle\right|^2\right] \leq \frac{\sqrt{3}\pi^2 \cdot \|\boldsymbol{y}\|_2^2}{d} \cdot \frac{1}{4^b}$ for any $b \geq 0$.
- For small bit-widths the above distortion upper bound can be further refined. Specifically, for $b = 1, 2, 3, 4$ we have $D_{\texttt{prod}}(Q_{\texttt{prod}}) \approx \frac{\mathbf{1.57}}{d}, \frac{\mathbf{0.56}}{d}, \frac{\mathbf{0.18}}{d}, \frac{\mathbf{0.047}}{d}$, respectively.

TurboQuant

- Also provide lower bounds

Lower Bound. In Theorem 3, we leverage Shannon's lower bound and Yao's minimax principle to prove that for any randomized quantization algorithm $Q : \mathbb{R}^d \to \{0,1\}^{b \cdot d}$ with bit-width b, there exist hard input instances $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ with $\|\boldsymbol{x}\| = 1$ such that the following lower bounds hold:

- $D_{\texttt{mse}}(Q) := \mathbb{E}\left[\left\|\boldsymbol{x} - Q^{-1}\left(Q(\boldsymbol{x})\right)\right\|_2^2\right] \geq \frac{1}{4^b}$
- $D_{\texttt{prod}}(Q) = \mathbb{E}\left[\left|\langle \boldsymbol{y}, \boldsymbol{x}\rangle - \langle \boldsymbol{y}, Q^{-1}\left(Q(\boldsymbol{x})\right)\rangle\right|^2\right] \geq \frac{\|\boldsymbol{y}\|_2^2}{d} \cdot \frac{1}{4^b}$

谢谢！

请提宝贵意见

中国科学技术大学
University of Science and Technology of China