Problem Set 4: Dequantizing QSVT

July 27, 2023

Before you begin, recall the definitions of sampling and query access for vectors and matrices ($\mathrm{SQ}(v), \mathrm{SQ}(A)$) and *oversampling* and query access ($\mathrm{SQ}_\phi(v), \mathrm{SQ}_\phi(A)$) [CGLLTW22, Definition 3.2]. Below, time complexities are in the word RAM model: basically, assume that reading input numbers, and performing operations on those numbers, cost $\mathcal{O}(1)$.

Problem 1 (Errare humanum est...)**.** Suppose we have $\mathrm{SQ}_{\phi_u}(u), \mathrm{SQ}_{\phi_v}(v)$ for vectors u, v. Show that we have $\mathrm{SQ}_\phi(A)$ for $A := uv^\dagger$ and $\phi = \phi_u\phi_v$ with cost $\mathbf{sq}_\phi(A) = \mathbf{sq}_{\phi_u}(u) + \mathbf{sq}_{\phi_u}(v)$.

Problem 2 (...sed perseverare (non?) diabolicum.)**.** Suppose we are given a matrix $A \in \mathbb{C}^{m\times m}$ with at most s non-zero entries per row, and suppose all entries are bounded by c. We are given this matrix as a list of non-zero entries $(i, j, A(i, j))$. Show how to perform $\mathrm{SQ}_\phi(A)$ queries for $\phi = c^2 \frac{sm}{\|A\|_F^2}$ with $\mathbf{sq}_\phi(A) = s$.[1] This means that we can run "dequantized" algorithms on sparse matrices with condition number κ; why doesn't this imply that QSVT admits no exponential speedup for sparse matrices?

Problem 3 (The alias method [Vos91])**.** Let $p = (p_1, \dots, p_m)$ be a set of probabilities, so $p_i \geq 0$ and $\sum p_i = 1$. Suppose also that all of the p_i's are described in binary with $\mathcal{O}(1)$ bits.

1. Suppose we are given a uniformly random number $x \in [0, 1]$ as a stream of random bits. Show how to sample $i \in [m]$ such that $\Pr[\text{sample } \ell] = p_\ell$ in $\mathcal{O}(m)$ operations.
2. Suppose we are given $p = (p_1, \dots, p_m)$ in the following form: we get a list of m probability distributions $d_1, \dots, d_m$ such that $\frac{1}{m}(d_1 + \dots + d_m) = p$ and every d_i is supported on at most two outcomes. Show that we can sample $i \in [m]$ according to p in $\mathcal{O}(1)$ time.
3. Prove that we can convert any distribution p into the form described above. Prove that we can do this in $\mathcal{O}(m)$ time.[2]

References

[CGLLTW22] Nai-Hui Chia, András Pal Gilyén, Tongyang Li, Han-Hsuan Lin, Ewin Tang, and Chunhao Wang. "Sampling-based sublinear low-rank matrix arithmetic framework for dequantizing quantum machine learning". In: *Journal of the ACM* 69.5 (Oct. 2022), pp. 1–72. DOI: 10.1145/3549524. arXiv: 1910.06151 [cs.DS] (page 1).

[Vos91] Michael D. Vose. "A linear algorithm for generating random numbers with a given distribution". In: *IEEE Transactions on Software Engineering* 17.9 (1991), pp. 972–975. DOI: 10.1109/32.92917 (page 1).

[1] Hint: We immediately have query access to A. What's a good upper bound that's easy to sample from?

[2] This implies that, if we get time to pre-process, we can get a data structure such that we can respond to $\mathrm{SQ}(v)$ queries in $\mathcal{O}(1)$ time (in the word RAM access model).