EE6367: Topics in Data Storage and Communications **2023**

Lecture 8: 21 September 2023

Instructor: Shashank Vatedka *Scribe: Gautam Singh*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

8.1 Vector Quantization Using Shared Randomness

Suppose that $\mathbf{X} \sim \text{iid}\left(\mathcal{N}\left(\mathbf{0}, \beta^2\right)\right)$. Then,

$$\mathbb{E}\left[\|\mathbf{X}\|^2\right] = \mathbb{E}\left[\sum_{i=1}^{d} X_i^2\right] \tag{8.1}$$
$$= d\beta^2. \tag{8.2}$$

Further, we note that

$$\Pr\left(\|\mathbf{X}\|^2 \geqslant d\beta^2\left(1+\delta\right)\right) \leqslant e^{-\theta(d)} \tag{8.3}$$

Consider a class of algorithms where the input vector is rotated using a uniformly chosen rotation matrix $\mathbf{R} \in \mathbb{M}^{d\times d}$ before encoding and then rotated back after decoding by applying $\mathbf{R}^{-1}$. Clearly, $\mathbf{R}\mathbf{R}^\top = \mathbf{I}$, and for an encoded vector $\mathbf{x}$, $\mathbf{y} \triangleq \mathbf{R}\mathbf{x}$ is also similarly distributed to $\mathbf{x}$. Since $\mathbf{x}$ is isotropically distributed, that is, depends on its 2-norm, it follows that

$$\frac{\mathbf{X}}{\|\mathbf{X}\|} \sim \text{Unif}\left(\mathbb{S}\left(\mathbf{0}, 1\right)\right). \tag{8.4}$$

For these schemes, a careful analysis shows

$$\text{Cost} \leqslant \theta\left(\frac{\|\mathbf{x}\|^2 \log d}{2^\alpha}\right). \tag{8.5}$$

However, a small tweak to the above scheme can lead to better performance. Assume that $k = d\left(1 + \mathcal{O}1\right)$. The encoder takes a random rotation vector $\mathbf{R}$ and computes $\mathbf{y} = \mathbf{R}\mathbf{x}$, and the vector $\mathbf{c}$ is transmitted, where

$$c_i \triangleq \text{sign}\left(y_i\right) = \begin{cases} 1 & y_i \geqslant 0 \\ -1 & \text{otherwise} \end{cases}. \tag{8.6}$$

Along with $\mathbf{c}$, a scale factor $s \in \mathbb{R}$ is also transmitted.

Given $\mathbf{c}$ and s, the decoder computes

$$\hat{\mathbf{X}} \triangleq s\mathbf{R}^\top \mathbf{c}. \tag{8.7}$$

The most computationally intensive part is that of multiplication at the encoder, which gives an overall time compelxity of $\mathcal{O}\left(d^2\right)$.

The squared error of this scheme is given by

$$\begin{aligned}
\left\|\mathbf{x} - \hat{\mathbf{X}}\right\|_2^2 &= \left\|\mathbf{R}\left(\mathbf{x} - \hat{\mathbf{X}}\right)\right\|_2^2 && (8.8)\\
&= \|\mathbf{R}\mathbf{x}\|_2^2 + \left\|\mathbf{R}\hat{\mathbf{X}}\right\|_2^2 - 2 < \mathbf{R}\mathbf{x}, \mathbf{R}\hat{\mathbf{X}} > && (8.9)\\
&= \|\mathbf{x}\|_2^2 + s^2 \|\mathbf{c}\|_2^2 - 2 < \mathbf{R}\mathbf{x}, s\mathbf{c} > && (8.10)\\
&= \|\mathbf{x}\|_2^2 + s^2 d - 2s \sum_{i=1}^{d} |\mathbf{R}\mathbf{x}_i| && (8.11)\\
&= \|\mathbf{x}\|_2^2 + s^2 d - 2s \|\mathbf{R}\mathbf{x}\|_1 && (8.12)\\
&\geqslant \|\mathbf{x}\|_2^2 - \frac{\|\mathbf{R}\mathbf{x}\|_1^2}{d} = \theta\|\mathbf{x}\|_2^2 && (8.13)
\end{aligned}$$

where the minimum is achieved at $s_{\min} = \frac{\|\mathbf{R}\mathbf{x}\|_1}{d}$.