COMPSCI 514: Algorithms for Data Science

Cameron Musco

University of Massachusetts Amherst. Fall 2026.

Lecture 14

In-Class Quiz 6

Question 1: TBD

Question 2: TBD

Logistics

- Problem Set 3 was posted last Friday. Due next Friday 11/13.
- Midterm 2 is Wednesday 11/18.
- Problem Set 2 grades were a bit delayed. Grading is done, just needs review. Out this week.

Summary

Last Class:

- Wrap up analysis of LSH table/hash function repetitions.
- Problem 1 of the problem set will do a more concrete theoretical analysis.
- Start on the JL Lemma for compressing high dimension vectors while preserving distances.

Today:

- Finish up proof of the JL Lemma.
- Application of JL to vector quantization for fast attention/near neighbor search.

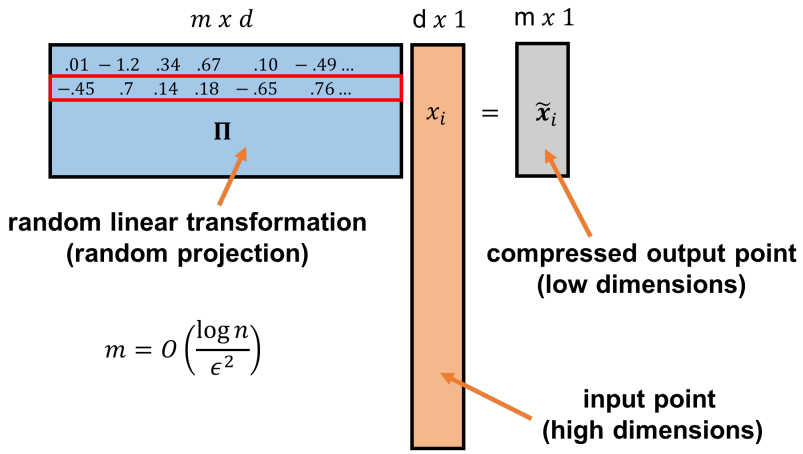
Recap: The Johnson-Lindenstrauss Lemma

For any $\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$, if we let $m = O\left(\frac{\log(n/\delta)}{\epsilon^2}\right)$ and choose

$$\mathbf{\Pi} \in \mathbb{R}^{m \times d}, \qquad \mathbf{\Pi}_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1/m),$$

with probability at least $1 - \delta$, every pairwise distance satisfies

$$(1-\epsilon)\|\vec{x}_i - \vec{x}_j\|_2 \leq \|\mathbf{\Pi}\vec{x}_i - \mathbf{\Pi}\vec{x}_j\|_2 \leq (1+\epsilon)\|\vec{x}_i - \vec{x}_j\|_2.$$



The target dimension depends on n and the error, not on d.

Recap: The Distributional JL Lemma

We showed that the standard JL Lemma follows from a closely related lemma:

Distributional JL Lemma: Let $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$ have each entry chosen i.i.d. as $\mathcal{N}(0, 1/m)$. If we set $m = O\left(\frac{\log(1/\delta)}{\epsilon^2}\right)$, then for any $\vec{y} \in \mathbb{R}^d$, with probability $\geq 1 - \delta$

$$(1-\epsilon)\|\vec{y}\|_2 \leq \|\mathbf{\Pi}\vec{y}\|_2 \leq (1+\epsilon)\|\vec{y}\|_2$$

I.e., Applying a random matrix $\mathbf{\Pi}$ to any vector $\vec{y}$ preserves $\vec{y}$'s norm with high probability.

- JL Lemma follows by applying to each $\vec{y}_{ij} = \vec{x}_i - \vec{x}_j$ and taking a union bound over these $\binom{n}{2}$ vectors.
- Use failure probability $\delta' = \delta/\binom{n}{2}$ so $\log(1/\delta') = O(\log(n/\delta))$.

$\mathbf{\Pi} \in \mathbb{R}^{m \times d}$: random projection matrix. d: original dimension. m: compressed dimension, ϵ: embedding error, δ: embedding failure prob.

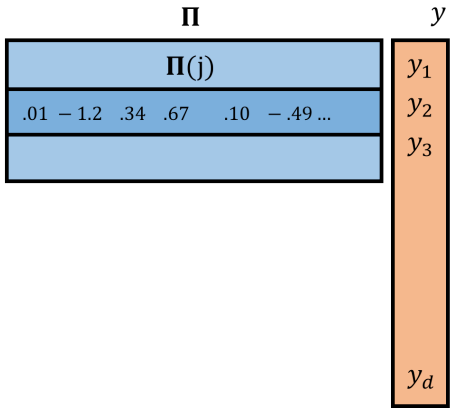
Proof of the Distribution JL Lemma

Distributional JL Proof

Distributional JL Lemma: Let $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$ have each entry chosen i.i.d. as $\mathcal{N}(0, 1/m)$. If we set $m = O\left(\frac{\log(1/\delta)}{\epsilon^2}\right)$, then for any $\vec{y} \in \mathbb{R}^d$, with probability $\geq 1 - \delta$

$$(1-\epsilon)\|\vec{y}\|_2 \leq \|\mathbf{\Pi}\vec{y}\|_2 \leq (1+\epsilon)\|\vec{y}\|_2$$

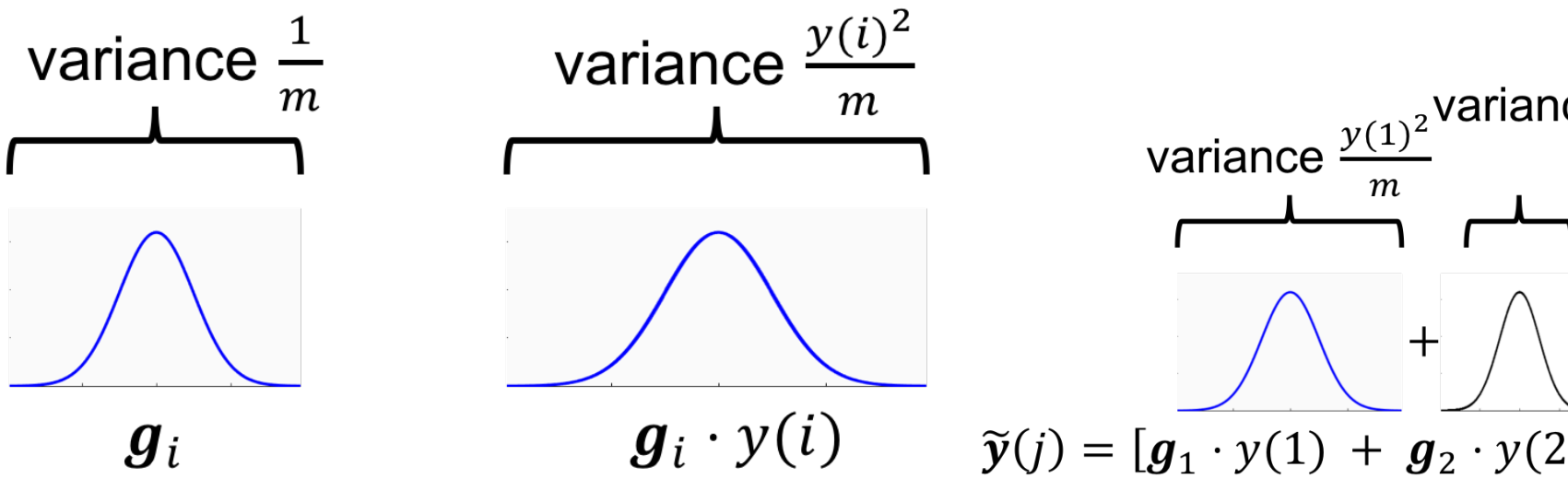
- Let $\tilde{\mathbf{y}}$ denote $\mathbf{\Pi}\vec{y}$ and let $\mathbf{\Pi}(j)$ denote the j^{th} row of $\mathbf{\Pi}$.
- For any j, $\tilde{\mathbf{y}}(j) = \langle \mathbf{\Pi}(j), \vec{y} \rangle = \sum_{i=1}^{d} \mathbf{g}_i \cdot \vec{y}(i)$ where $\mathbf{g}_i \sim \mathcal{N}(0, 1/m)$.



$\vec{y} \in \mathbb{R}^d$: arbitrary vector, $\tilde{\mathbf{y}} \in \mathbb{R}^m$: compressed vector, $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$: random projection. d: original dim. m: compressed dim, ϵ: error, δ: failure prob.

Distributional JL Proof

- Let $\tilde{\mathbf{y}}$ denote $\mathbf{\Pi}\vec{y}$ and let $\mathbf{\Pi}(j)$ denote the j^{th} row of $\mathbf{\Pi}$.
- For any j, $\tilde{\mathbf{y}}(j) = \langle \mathbf{\Pi}(j), \vec{y} \rangle = \sum_{i=1}^{d} \mathbf{g}_i \cdot \vec{y}(i)$ where $\mathbf{g}_i \sim \mathcal{N}(0, 1/m)$.
- $\mathbf{g}_i \cdot \vec{y}(i) \sim \mathcal{N}(0, \frac{\vec{y}(i)^2}{m})$: normally distributed with variance $\frac{\vec{y}(i)^2}{m}$.



What is the distribution of $\tilde{\mathbf{y}}(j)$? Also Gaussian!

$\vec{y} \in \mathbb{R}^d$: arbitrary vector, $\tilde{\mathbf{y}} \in \mathbb{R}^m$: compressed vector, $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$: random projection mapping $\vec{y} \to \tilde{\mathbf{y}}$. $\mathbf{\Pi}(j)$: j^{th} row of $\mathbf{\Pi}$, d: original dimension. m: compressed dimension, $\mathbf{g}_i$: normally distributed random variable.

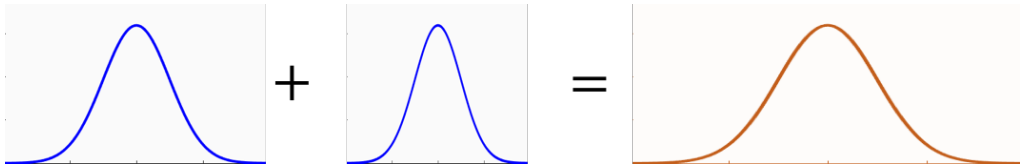
Distributional JL Proof

Letting $\tilde{\mathbf{y}} = \mathbf{\Pi}\vec{y}$, we have $\tilde{\mathbf{y}}(j) = \langle \mathbf{\Pi}(j), \vec{y} \rangle$ and:

$$\tilde{\mathbf{y}}(j) = \sum_{i=1}^{d} \mathbf{g}_i \cdot \vec{y}(i) \text{ where } \mathbf{g}_i \cdot \vec{y}(i) \sim \mathcal{N}\left(0, \frac{\vec{y}(i)^2}{m}\right).$$

Stability of Gaussian Random Variables. For independent $a \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $b \sim \mathcal{N}(\mu_2, \sigma_2^2)$ we have:

$$a + b \sim \mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$$



Thus, $\tilde{\mathbf{y}}(j) \sim \mathcal{N}(0, \frac{\vec{y}(1)^2}{m} + \frac{\vec{y}(2)^2}{m} + \ldots + \frac{\vec{y}(d)^2}{m} \frac{\|\vec{y}\|_2^2}{m})$ I.e., $\tilde{\mathbf{y}}$ itself is a random Gaussian vector. Rotational invariance of the Gaussian distribution. Stability is another explanation for the central limit theorem.

$\vec{y} \in \mathbb{R}^d$: arbitrary vector, $\tilde{\mathbf{y}} \in \mathbb{R}^m$: compressed vector, $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$: random

Distributional JL Proof

So far: Letting $\Pi \in \mathbb{R}^{d \times m}$ have each entry chosen i.i.d. as $\mathcal{N}(0, 1/m)$, for any $\vec{y} \in \mathbb{R}^d$, letting $\tilde{\mathbf{y}} = \Pi\vec{y}$:

$$\tilde{\mathbf{y}}(j) \sim \mathcal{N}(0, \|\vec{y}\|_2^2/m).$$

What is $\mathbb{E}[\|\tilde{\mathbf{y}}\|_2^2]$?

$$\mathbb{E}[\|\tilde{\mathbf{y}}\|_2^2] = \mathbb{E}\left[\sum_{j=1}^{m} \tilde{\mathbf{y}}(j)^2\right] = \sum_{j=1}^{m} \mathbb{E}[\tilde{\mathbf{y}}(j)^2]$$

$$= \sum_{j=1}^{m} \frac{\|\vec{y}\|_2^2}{m} = \|\vec{y}\|_2^2$$

So $\tilde{\mathbf{y}}$ has the right norm in expectation.

How is $\|\tilde{\mathbf{y}}\|_2^2$ distributed? Does it concentrate?

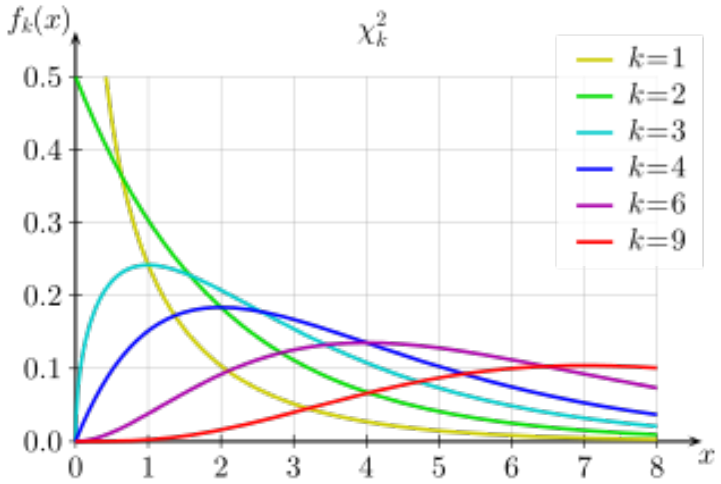
$\vec{y} \in \mathbb{R}^d$: arbitrary vector, $\tilde{\mathbf{y}} \in \mathbb{R}^m$: compressed vector, $\Pi \in \mathbb{R}^{m \times d}$: random projection mapping $\vec{y} \to \tilde{\mathbf{y}}$. $\Pi(j)$: j^{th} row of Π, d: original dimension. m: compressed dimension, $\mathbf{g}_i$: normally distributed random variable

Distributional JL Proof

So far: Letting $\mathbf{\Pi} \in \mathbb{R}^{d \times m}$ have each entry chosen i.i.d. as $\mathcal{N}(0, 1/m)$, for any $\vec{y} \in \mathbb{R}^d$, letting $\tilde{\mathbf{y}} = \mathbf{\Pi}\vec{y}$:

$$\tilde{\mathbf{y}}(j) \sim \mathcal{N}(0, \|\vec{y}\|_2^2/m) \text{ and } \mathbb{E}[\|\tilde{\mathbf{y}}\|_2^2] = \|\vec{y}\|_2^2$$

$\|\tilde{\mathbf{y}}\|_2^2 = \sum_{i=1}^{m} \tilde{\mathbf{y}}(j)^2$ a Chi-Squared random variable with m degrees of freedom (a sum of m squared independent Gaussians)

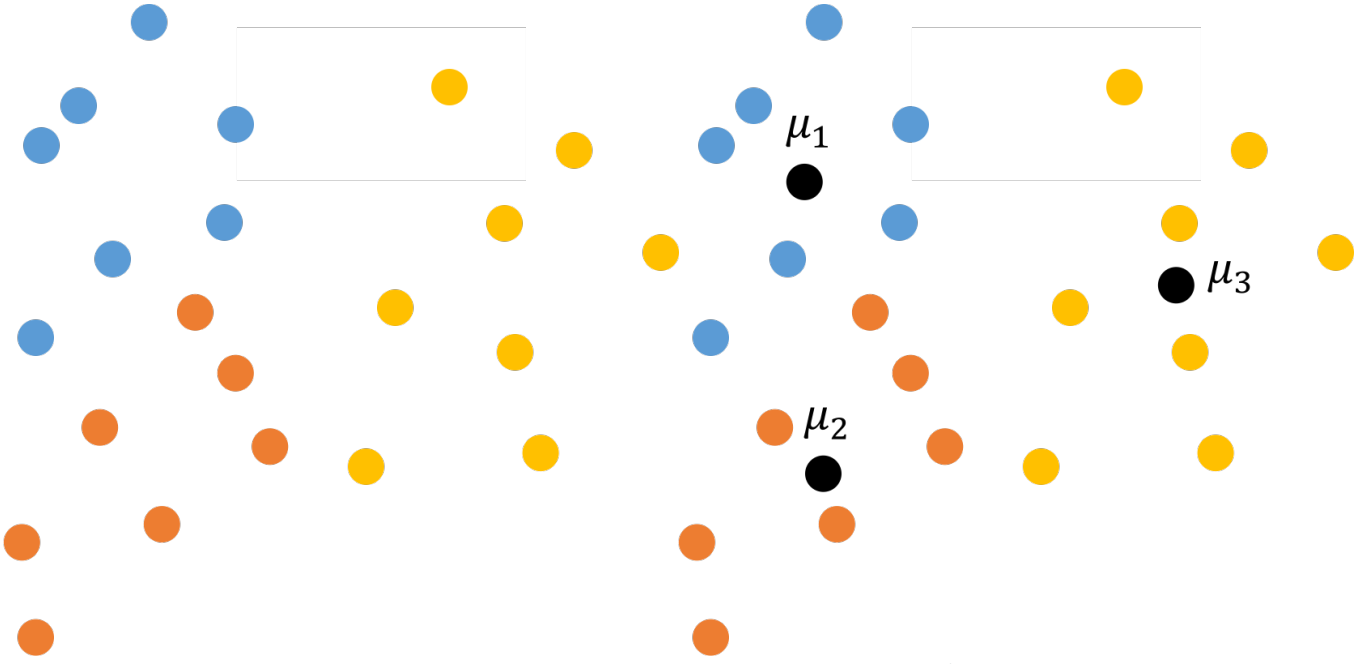


Lemma: (Chi-Squared Concentration) Letting $\mathbf{Z}$ be a Chi-Squared random variable with m degrees of freedom,

$$\Pr[|\mathbf{Z} - \mathbb{E}\mathbf{Z}| > \epsilon\mathbb{E}\mathbf{Z}] < 2e^{-m\epsilon^2/8}.$$

Example Application: k-means clustering

Goal: Separate n points in d dimensional space into k groups.



k-means Objective: $Cost(\mathcal{C}_1, \ldots, \mathcal{C}_k) = \min_{\mathcal{C}_1, \ldots \mathcal{C}_k} \sum_{j=1}^{k} \sum_{\vec{x} \in \mathcal{C}_k} \|\vec{x} - \mu_j\|_2^2.$

Write in terms of distances (will work through on Pset 3):

$$Cost(\mathcal{C}_1, \ldots, \mathcal{C}_k) = \min_{\mathcal{C}_1, \ldots \mathcal{C}_k} \sum_{j=1}^{k} \sum_{\vec{x}_1, \vec{x}_2 \in \mathcal{C}_k} \|\vec{x}_1 - \vec{x}_2\|_2^2$$

Example Application: k-means clustering

k-means Objective: $Cost(\mathcal{C}_1, \ldots, \mathcal{C}_k) = \min_{\mathcal{C}_1, \ldots \mathcal{C}_k} \sum_{j=1}^{k} \sum_{\vec{x}_1, \vec{x}_2 \in \mathcal{C}_k} \|\vec{x}_1 - \vec{x}_2\|_2^2$

If we randomly project to $m = O\left(\frac{\log n}{\epsilon^2}\right)$ dimensions, for all pairs $\vec{x}_1, \vec{x}_2$,

$$(1-\epsilon)\|\vec{x}_1 - \vec{x}_2\|_2^2 \leq \|\tilde{\mathbf{x}}_1 - \tilde{\mathbf{x}}_2\|_2^2 \leq (1+\epsilon)\|\vec{x}_1 - \vec{x}_2\|_2^2 \implies$$

Letting $\overline{Cost}(\mathcal{C}_1, \ldots, \mathcal{C}_k) = \min_{\mathcal{C}_1, \ldots \mathcal{C}_k} \sum_{j=1}^{k} \sum_{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2 \in \mathcal{C}_k} \|\tilde{\mathbf{x}}_1 - \tilde{\mathbf{x}}_2\|_2^2$

$$(1-\epsilon)Cost(\mathcal{C}_1, \ldots, \mathcal{C}_k) \leq \overline{Cost}(\mathcal{C}_1, \ldots, \mathcal{C}_k) \leq (1+\epsilon)Cost(\mathcal{C}_1, \ldots, \mathcal{C}_k).$$

Upshot: Can cluster in m dimensional space (more efficient) and minimize $\overline{Cost}(\mathcal{C}_1, \ldots, \mathcal{C}_k)$. The optimal set of clusters will have true cost within $1 + c\epsilon$ times the true optimal. See Pset 3 for full analysis.

Applications of JL to Vector Quantization

Vector Quantization

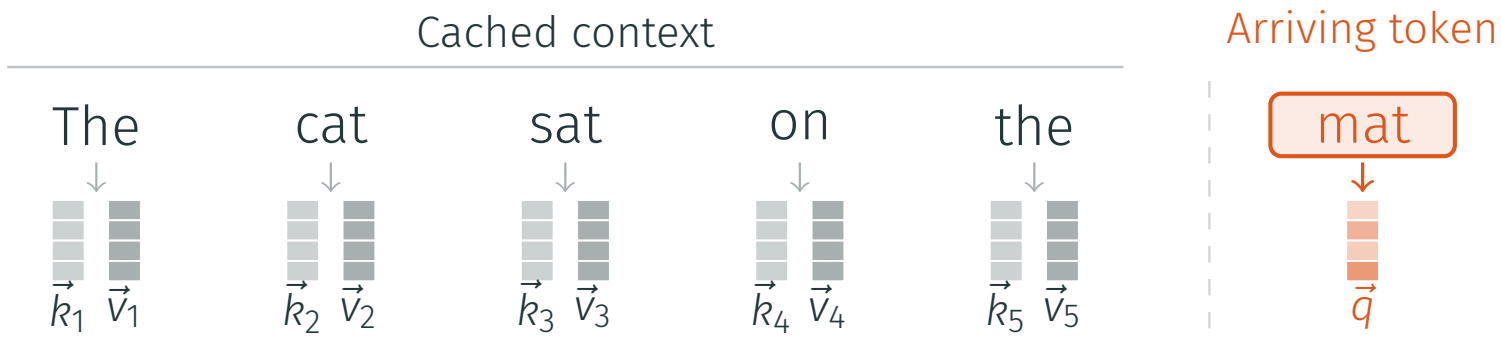
Suppose we have many stored vectors $\vec{k}_1, \ldots, \vec{k}_n \in \mathbb{R}^d$ and receive queries $\vec{q} \in \mathbb{R}^d$.

Goal: Estimate $\langle \vec{q}, \vec{k}_i \rangle$ for each stored vector while minimizing space usage/query time.

Vector quantization: Represent each $\vec{k}_i$ with a compressed version, at the cost of only giving an approximate inner product.

- JL random projection gives a natural solution to this problem. Instead of storing $\vec{k}_1, \ldots, \vec{k}_n \in \mathbb{R}^d$, store $\mathbf{\Pi}\vec{k}_1, \ldots, \mathbf{\Pi}\vec{k}_n \in \mathbb{R}^m$.
- Estimate $\langle \vec{q}, \vec{k}_i \rangle \approx \langle \mathbf{\Pi}\vec{q}, \mathbf{\Pi}\vec{k}_i \rangle$.
- Requires $b \cdot m$ bits per vector, where b is the number of bits used per entry.
- How big does b need to be? Can we compress it somehow?

Motivation: The Key-Value Cache in LLMs



The current query $\vec{q}$ attends to the cached keys and values:

$$p_i = \mathrm{softmax}\left(\frac{\langle \vec{q}, \vec{k}_i \rangle}{\sqrt{d}}\right), \qquad \vec{o} = \sum_i p_i \vec{v}_i.$$

- The key-value cache stores previous keys and values to avoid recomputing them for each new token.
- Contexts can have nearly a million tokens, motivating compression of this cache using vector quantization.