Linear dimension reduction via Johnson-Lindenstrauss

To reduce computational complexity and memory, we want to map a high dimensional data set onto a lower-dimensional space while preserving important structure.

What kind of structure can we preserve in that case? One natural choice is to try to (approximately) preserve distance between points.

Suppose we are given n points $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ in $\mathbb{R}^d$ where n, d are large. We want to find points $\boldsymbol{y}_1, \dots, \boldsymbol{y}_n$ in $\mathbb{R}^k$ with $k \ll d$ such that $\|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2 \approx \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2$ for all pairs $i, j = 1, \dots, n$. If $d > n$ we can always choose $k = n$ and project $\boldsymbol{x}_i \in \mathbb{R}^d$ onto $\mathbb{R}^n$ and preserve distances exactly, but we want $k \ll d$. We will look for a linear map $f : \mathbb{R}^d \to \mathbb{R}^k$ such that

$$(1-\varepsilon)\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 \le \|f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)\|_2^2 \le (1+\varepsilon)\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2,$$

for all pairs $i, j = 1, \dots, n$, and for any $\varepsilon > 0$. We call such a mapping f an ε-isometry.

- For which k and which ε does such an f exist?
- How do we construct f?
- Can we construct, store, and apply f numerically efficiently?

Random projections will address the first two questions. “Structured” random projections will also take care of the last question.

Theorem 1 (Johnson-Lindenstrauss Lemma [3]). *For any $0 < \varepsilon < 1$ and any integer n, let k be such that*

$$k \ge \frac{4 + 2\alpha}{\frac{\varepsilon^2}{2} - \frac{\varepsilon^3}{3}} \log n,$$

for some $\alpha > 0$. Then for any arbitrary set of points $\boldsymbol{x}_1, \dots \boldsymbol{x}_n \in \mathbb{R}^d$ with probability at least $1 - n^{-\alpha}$ there exists a linear map $f : \mathbb{R}^d \to \mathbb{R}^k$ such that for all $i, j = 1, \dots, n$:

$$(1-\varepsilon)\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 \le \|f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)\|_2^2 \le (1+\varepsilon)\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2, \tag{1}$$

Our proof follows Dasgupta and Gupta [2]. We will make use of the following lemma.

Lemma 2 ([2]). *Let $z_1, \dots, z_d$ be iid standard (mean 0, variance 1) Gaussian random variables and $\boldsymbol{z} = [z_1, \dots, z_d]$. Let g be the projection onto the first k coordinates of $\boldsymbol{z}$ and*

$$\boldsymbol{y} = g\left(\frac{\boldsymbol{z}}{\|\boldsymbol{z}\|_2}\right) = \frac{1}{\|\boldsymbol{z}\|_2}[z_1, \dots, z_k].$$

Denote $L = \|\boldsymbol{y}\|_2^2$. Then $\mathbb{E}\, L = \frac{k}{d}$ and

- *If $\delta < 1$: $\mathbb{P}\left(L \le \delta \frac{k}{d}\right) \le \exp\left(\frac{k}{2}(1 - \delta + \log \delta)\right)$.*

- *If $\delta > 1$:* $\mathbb{P}\left(L \geq \delta\frac{k}{d}\right) \leq \exp\left(\frac{k}{2}(1-\delta+\log\delta)\right)$.

Proof of JL:. Key idea: We will construct a projection onto a random subspace of dimension k which will satisfy the ε-isometry property with a certain probability.

What is a k-dimensional random subspace? Pick a random line through the origin in $\mathbb{R}^d$, pick a second random line through the origin orthogonal to the first, and so on.

However, it is difficult to work with random subspaces for proof purposes.

Observation: Projecting a fixed vector onto a random subspace is the same as projecting a random vector of fixed length onto a fixed subspace. That is true since one can rotate the coordinate system so that a set of basis vectors for the random subspace are the first k coordinate axes.

Due to homogeneity of (1) we can assume without loss of generality that $\boldsymbol{x}_i - \boldsymbol{x}_j$ has unit-norm, so $\boldsymbol{x}_i - \boldsymbol{x}_j \in S^{d-1}$ (the unit sphere in $\mathbb{R}^d$), We can generate a random vector in S^{d-1} by taking a standard Gaussian random variable in $\mathbb{R}^d$ and normalizing it to unit-norm.

Let $g : \mathbb{R}^d \to \mathbb{R}^k$ be the projection onto k-dimensional random subspace and let $f : \mathbb{R}^d \to \mathbb{R}^k$ be given by $\frac{d}{k}g$.

Then by the observation above $\frac{\|f(\boldsymbol{x}_i)-f(\boldsymbol{x}_j)\|_2^2}{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2^2}$ has the same distribution as $\frac{d}{k}L$ in Lemma 2. Thus

$$\mathbb{P}\left(\frac{\|f(\boldsymbol{x}_i)-f(\boldsymbol{x}_j)\|_2^2}{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2^2} \leq 1-\varepsilon\right) \leq \exp\left(\frac{k}{2}(1-(1-\varepsilon)+\log(1-\varepsilon))\right).$$

Note that for $\varepsilon \geq 0$, $\log(1-\varepsilon) \leq -\varepsilon - \frac{\varepsilon^2}{2}$. Thus we have the following inequalities:

$$\mathbb{P}\left(\frac{\|f(\boldsymbol{x}_i)-f(\boldsymbol{x}_j)\|_2^2}{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2^2} \leq 1-\varepsilon\right) \leq \exp\left(-\frac{k\varepsilon^2}{4}\right) \leq \exp(-2\log n) = \frac{1}{n^{2+\alpha}}.$$

$$\mathbb{P}\left(\frac{\|f(\boldsymbol{x}_i)-f(\boldsymbol{x}_j)\|_2^2}{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2^2} \geq 1+\varepsilon\right) \leq \exp(-2\log n) = \frac{1}{n^{2+\alpha}}.$$

Using the union bound, we get that for fixed i and j there holds

$$\mathbb{P}\left(\frac{\|f(\boldsymbol{x}_i)-f(\boldsymbol{x}_j)\|_2^2}{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2^2} \notin [1-\varepsilon, 1+\varepsilon]\right) \leq \frac{2}{n^{2+\alpha}}.$$

This is for one *fixed* pair $\boldsymbol{x}_i, \boldsymbol{x}_j$. But we have $\binom{n}{2}$ such pairs. Using the union bound over all $\binom{n}{2}$ such pairs yields

$$\mathbb{P}\left(\exists\,(i,j) : \frac{\|f(\boldsymbol{x}_i)-f(\boldsymbol{x}_j)\|_2^2}{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2^2} \notin [1-\varepsilon, 1+\varepsilon]\right) \leq \frac{2}{n^{2+\alpha}}\frac{n(n-1)}{2} = \frac{1}{n^\alpha}(1-\frac{1}{n}) \leq \frac{1}{n^\alpha}.$$

We conclude that f has the desired property (1) with probability at least $1 - \frac{1}{n^\alpha}$.

□

The Johnson-Lindenstrauss projection theorem holds for all $\boldsymbol{x}_i$, $\boldsymbol{x}_j$, $i, j = 1, ..., n$, and not just for most x_i, x_j. Unless n is exponentially larger than d, the relation $k \sim \frac{\log n}{\varepsilon^2}$ gives $k < d$, therefore we can consider this a dimension reduction.

JL can be applied in a "streaming" manner. We can construct $\boldsymbol{P}$ beforehand; all we need to know is (an estimate of) the size n. We do not need to wait until all vectors x_i are available, instead we can start computing $\boldsymbol{P}\boldsymbol{x}_j$ as the vectors $\boldsymbol{x}_i$ arrive. In that sense JL is essentially a data-agnostic method as $\boldsymbol{P}$ only depends on n and d, but not on the specific vectors $\boldsymbol{x}_i$. This is in sharp contrast to another linear dimension reduction method, namely *Principal Component Analysis*, which is fully data dependent. It depends on the application whether data dependency in the dimension reduction is an advantage and desirable or not.

How can we construct the mapping f? We can choose for example a Gaussian random matrix $\boldsymbol{A}$ of size $k \times d$. [`Matlab: >>randn(k,d)`] We simply set $\boldsymbol{P} := \sqrt{\frac{1}{k}}\boldsymbol{A}$. The downsides of this approach (and Johnson-Lindenstrauss in general) are the cost for storing and applying $\boldsymbol{P}$.

0.1 Fast Johnson-Lindenstrauss Projection

Can we construct a more efficient version of the Johnson-Lindenstrauss random projection? Answering this question leads us to the Fast Johnson-Lindenstrauss random projection, of which there are numerous versions, see e.g. [1, 4]. This brings up the question, whether we could simply (randomly) sample entries of $\boldsymbol{x} \in \mathbb{R}^d$ to reduce its dimension? Call such a sampling matrix $\boldsymbol{S}$: $\boldsymbol{S}$ has size $k \times d$. Each row of S has a single non-zero entry $1 \cdot \sqrt{\frac{d}{k}}$ at a uniformly random location. For any $\boldsymbol{x} \in \mathbb{R}^d$

$$\begin{aligned} \mathbb{E}[(\boldsymbol{S}\boldsymbol{x})_i^2] &= \sum_{j=1}^{d} \mathbb{P}(\text{random location is } j) \cdot \frac{d}{k} \cdot x_j^2 = \frac{1}{k}\|\boldsymbol{x}\|_2^2 \\ \Rightarrow \quad \mathbb{E}[\|\boldsymbol{S}\boldsymbol{x}\|_2^2] &= \mathbb{E}[\sum_{i=1}^{k}(\boldsymbol{S}\boldsymbol{x})_i^2] = \|\boldsymbol{x}\|_2^2 \end{aligned}$$

In expectation this choice is ok, but the variance of $||\boldsymbol{S}\boldsymbol{x}||_2^2$ is bad! If for instance $\boldsymbol{x}$ has only one non-zero entry, then we need $k \sim O(d)$ to ensure that $\|\boldsymbol{S}\boldsymbol{x}\|_2^2 \neq 0$. Hence if we want to apply random subsampling, we would need to avoid sparse vectors. This suggests to first apply a random rotation so that sparse vectors become non-sparse. (This could in turn transform non-sparse vectors into sparse ones, but fortunately the probability for this is exceedingly small). This random rotation idea is already present in the standard Johnson-Lindenstrauss approach.

It is instructive to compare the sensitivity of different norms to sparsity or non-sparsity, respectively. Consider

$$\boldsymbol{u} = [1, 0, 0, ..., 0] \in \mathbb{R}^d : \|\boldsymbol{u}\|_2 = 1, \|\boldsymbol{u}\|_\infty = 1$$

$$\boldsymbol{u} = [1, 1, 1, ..., 1]\frac{1}{\sqrt{d}} \in \mathbb{R}^d : \|\boldsymbol{u}\|_2 = 1, \|\boldsymbol{u}\|_\infty = \frac{1}{\sqrt{d}}$$

Hence, our goal is now clear. We should employ a rotation matrix (a unitary matrix) $\boldsymbol{R}$ which maps vectors $\boldsymbol{x}$ into vectors $\boldsymbol{Rx}$ such that

$$\frac{||\boldsymbol{Rx}||_\infty}{||\boldsymbol{Rx}||_2} \sim \frac{1}{\sqrt{d}}.$$

At the same time, in accordance with our other goals, we must be able to efficiently generate, store, and apply this rotation matrix. What rotation matrix shall we use to that end? Two natural choices for this are:

1. The Discrete Fourier Transform;
2. the Hadamard matrix.

Both matrices work equally well, both can be applied in $O(d \log d))$ operations via FFT-type algorithms and the proofs for both are essentially identical [1]. We therefore will focus on only one of them, namely the Hadamard matrix, more specifically the Walsh-Hamdard matrix $\boldsymbol{H}_d$ which is defined recursively by:

$$\boldsymbol{H}_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \; \boldsymbol{H}_4 = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}, \cdots, \boldsymbol{H}_d = \frac{1}{\sqrt{2}}\begin{bmatrix} \boldsymbol{H}_{\frac{d}{2}} & \boldsymbol{H}_{\frac{d}{2}} \\ \boldsymbol{H}_{\frac{d}{2}} & -\boldsymbol{H}_{\frac{d}{2}} \end{bmatrix}.$$

If there is no confusion about the dimension we will set $\boldsymbol{H} := \boldsymbol{H}_d$. Note that $\boldsymbol{H}\boldsymbol{H}^* = \boldsymbol{H}^*\boldsymbol{H} = \boldsymbol{I}$ and $|\boldsymbol{H}_{ii}| = \frac{1}{\sqrt{d}}$. We have assumed that d is a power of two, otherwise we can always append zeros to the given vectors so that their length becomes a power of two.

We can compute the matrix-vector product $\boldsymbol{Hx}$ in $O(d \log d)$ operations and $\boldsymbol{H}$ can be efficiently "stored" (well, we never actually store $\boldsymbol{H}$ explicitly).

However, it is difficult (perhaps impossible?) to prove that for arbitrary vector $\boldsymbol{x}$ there holds $\frac{\|\boldsymbol{Hx}\|_\infty}{\|\boldsymbol{Hx}\|_2} \approx \frac{1}{\sqrt{d}}$. Another issue is that some vectors, such as the vector with contant entries (which is not sparse at all), are mapped into very sparse vectors by the Hadamard matrix, which is the opposite of what we wanted to achieve. The issue here is that vectors with constant (or nearly constant) entries do appear in practice. We can easily avoid both aforementioned problems by introducing a bit of randomness in our rotation matrix. The idea is to replace $\boldsymbol{H}$ by $\boldsymbol{HD}$, where $\boldsymbol{D}$ is diagonal matrix with entries $D_{ii} = \pm 1$ with probability $\frac{1}{2}$. Note that $\boldsymbol{HD}$ is still unitary: $(\boldsymbol{HD})^* (\boldsymbol{HD}) = \boldsymbol{I}$. And we can still apply $\boldsymbol{HD}$ fast. We call $\boldsymbol{HD}$ a random Hadamard Matrix.

Lemma 3. *Let $\boldsymbol{x} \in \mathbb{R}^d$, $x \neq 0$. Let $\boldsymbol{y} = \boldsymbol{HDx}$, where $\boldsymbol{HD}$ is a random Hadamard matrix. Then,*

$$\mathbb{P}\left[\frac{\|\boldsymbol{y}\|_\infty}{\|\boldsymbol{y}\|_2} \geq \sqrt{\frac{2\log(4d/\delta)}{d}} \right] \leq \frac{\delta}{2}, \quad \forall\, 0 < \delta < 1.$$

Proof: Uses Hoeffding Inequality.

We are now ready to define the Fast Johnson-Lindenstrauss random projection.

Theorem 4. *There exists a $k \times d$ random matrix $\boldsymbol{Q} := \boldsymbol{SHD}$ with $k = O(\log(\frac{d}{\delta})^2 \log(\frac{1}{\delta})/\varepsilon^2)$, where $\boldsymbol{S}$ is the $k \times d$ sampling matrix introduced earlier, $\boldsymbol{H}$ is a $d \times d$ Hadamard matrix and $\boldsymbol{D}$ is a $d \times d$ binary diagonal matrix, such that for each pair of vectors $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^d$:*

$$(1-\varepsilon)\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 \leq \|\boldsymbol{Q}\boldsymbol{x}_i - \boldsymbol{Q}\boldsymbol{x}_j)\|_2^2 \leq (1+\varepsilon)\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2$$

with probability at least $1 - \delta$.

The computation cost for applying $\boldsymbol{Q}$ is $O(d \log d)$. Further improvements to FJL can be found e.g. in [4].

References

[1] Nir Ailon and Bernard Chazelle. The fast Johnson-Lindenstrauss transform and approximate nearest neighbors. *SIAM J. Comput.*, 39(1):302–322, 2009.

[2] S. Dasgupta and A. Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures Algorithms*, 22(1):60–65, 2003.

[3] W. B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemp. Math.*, 26:189–206, 1984.

[4] Felix Krahmer and Rachel Ward. New and improved Johnson-Lindenstrauss embeddings via the Restricted Isometry Property. *SIAM J. Math. Anal.*, 43(3):1269–1281, 2011.