RANDOM PROJECTION AND JL LEMMA: AN INTRODUCTION

FABIO COPPINI

ABSTRACT. An introductory note on random projection in Machine Learning, complemented with the key mathematical results. We include the main tool known as Johnson-Linderstrauss Lemma and provides the elementary proof given by Dasgupsta and Gupsta in 1999 together with some historical references. Some emphasis is put on the algorithmic implementations, as well as the notion of quasi-orthogonality given by Kainen and Kůrková in 1993 together with (the idea) of their beautiful proof.

The end of this note includes many references to applications.

ML Reading Group LPSM: The content of this note was presented at the Machine Learning Reading Group in LPSM (Université de Paris and Sorbonne Université) on Friday, 4 December 2020.

1. Introduction

1.1. **Context.** Suppose we have a set S of n points in $\mathbb{R}^d$

$$S = \{X_1, \dots, X_n\} \subset \mathbb{R}^d$$

for some large $d \in \mathbb{N}$. We would like to understand if it is possible to construct an application from S to a lower dimensional space, i.e., a projection,

$$R : S \subset \mathbb{R}^d \to \mathbb{R}^k$$

for $k < d$, such that R preserves the geometry of S, up to some error $\varepsilon > 0$. We say that R preserves the geometry of the set S, if it satisfies a property similar to:

$$\|R(u) - R(v)\| \approx \|u - v\|, \quad \text{for all } u, v \in S,$$

which means that the pairwise distances are preserved.

Since $k < d$, the map R cannot be an isometry on $\mathbb{R}^d$ (i.e., $\|R(u)\| = \|u\|$). However, we can look for an approximate isometric embedding: let $\varepsilon > 0$, we ask whether it is possible to find $f : S \subset \mathbb{R}^d \to \mathbb{R}^k$ such that

$$(1-\varepsilon)\|u - v\|^2 \leqslant \|f(u) - f(v)\|^2 \leqslant (1+\varepsilon)\|u - v\|^2, \quad u, v \in S. \tag{1.1}$$

Provided that this is possible, we are interested in:

(1) the relationship between ε, n, d and k;
(2) how to construct such an f.

Date: December 7, 2020.

Spoiler 1.1. *The answer is yes and the proof provides another example of the power behind the probabilistic method: indeed, we will see that almost every **linear** random projection (in some suitable class) yields the desired property 1.1*

1.2. A quick look at applications.

The main reason behind random projection is *dimensionality reduction.* By passing from a high-dimensional space, $\mathbb{R}^d$ in our case but a separable Hilbert space could be considered as well, to a lower dimensional space with a precise error rate, we can, e.g., drastically reduce the computational cost of many algorithms while monitoring the accuracy lost.

In a nutshell, random projections are used for:

(1) reduce the curse of dimensionality (high dimensional input data, small number of observations with a lot of features, reduce overfitting, etc.);
(2) improve computational performances whenever the underlying algorithm relies on the geometry of the input data (e.g., clustering, nearest-neighbor, etc.) at the cost of accuracy;
(3) data compression, by representing (usually sparse) data in a lower dimensional space, thus reducing the computational cost in space;
(4) many other applications: matrix completion, compressed learning, density estimation... more on that in Section 3.

1.3. But why?

Random projection is supported by accurate mathematical results. Notably, the several implementations are well understood and the trade-off between computation cost and accuracy is explicit.

But there is more than that, namely:

- there are implementations for R which are fast and computationally cheap (see Fast JL Transform in Subsection 2.3);
- random projection is transparent and interpretable, i.e., the transformation R is explicit so that it is possible to recover the original *features*;
- it is distribution independent (e.g., contrary to Principal Component Analysis): we can easily analyze the data after projecting it.

2. Mathematics behind the scene and algorithm implementation

2.1. Key mathematical results.

The first mathematical result which proves the existence of a map f satisfying (1.1) was given by Johnson and Linderstrauss in 1984 in the paper *Extension of Lipschitz maps into a Hilbert space*, see [14].

The original statement is equivalent to the following one.

Theorem 2.1 (original JL Lemma [14, Lemma 1.1])**.** *For each $0 < \varepsilon < 1$, there exists a constant $K = K(\varepsilon) > 0$ so that if $S \subset \mathbb{R}^d$ with $|S| = n$ for some $n = 2, 3, \dots,$ there is a mapping $f : S \to \mathbb{R}^k$, with $k = [K \log n]$, which satisfies* (1.1)*, i.e., such that*

$$(1-\varepsilon)\left\|u-v\right\|^2 \leqslant \left\|f(u)-f(v)\right\|^2 \leqslant (1+\varepsilon)\left\|u-v\right\|^2, \quad u, v \in S. \tag{2.1}$$

A few remarks are in order here:

- The proof is probabilistic and rather involved (we skip it and refer to [14]!);
- The construction of f is implicit, i.e., not directly possible to be implemented;
- The constant k does not depend on d but only[1] on $n = |S|$ and ε!

In order to obtain a first constructive proof, we need to wait for the work of Frankl and Maehara [12]. Their work concerns the distribution of the angle between random vectors in finite-dimensional space (related to the following work by Kainen and Kůrková, see [16] and Subsection 2.2).

As a corollary of their main result, they give a proof of Theorem 2.1 based on projections on random k-dimensional hyperplanes. They show that the map f can be chosen among these random projections and that it is thus linear, i.e. $f(x) = Rx$, where R is a $k \times d$ real matrix.

This result has been further improved in the constants, by Dasgupta and Gupta [7] in 1999 (see also Indyk and Motwani [13]. We will follow their simple probabilistic proof.

Here one of the most common versions of JL Lemma.

Theorem 2.2 ([7, Theorem 2.1]). *For any $0 < \varepsilon < 1$ and any integer n, let k be a positive integer such that*

$$k \geqslant 4(\varepsilon^2/2 - \varepsilon^3/3)^{-1} \ln n. \tag{2.2}$$

Then for any set S of n points in $\mathbb{R}^d$, there is a linear map $R : \mathbb{R}^d \to \mathbb{R}^k$ such that for all $u, v \in S$ (1.1) holds, i.e.,

$$(1-\varepsilon)\left\|u-v\right\|^2 \leqslant \left\|Ru - Rv\right\|^2 \leqslant (1+\varepsilon)\left\|u-v\right\|^2, \quad u, v \in S.$$

Idea of the proof. Let f be a random projection on a k-dimensional subset of $\mathbb{R}^d$. Since f is linear, we can study $f(u)$ for some unit vector u. The theorem is proved by showing that the squared length of a random $u \in \mathbb{R}^d$ is concentrated about its mean when projected through f, and is not distorted by more than $(1 \pm \varepsilon)$ with probability $1 - n^{-2}$. The union bound on all possible couples in S gives that the probability that (1.1) holds, is bigger than $1 - 1/n$. This concludes the proof.

We thus want to prove that, for $u \in \mathbb{S}^{d-1}$ and f a projection onto a random k-dimensional subspace of $\mathbb{R}^d$, it holds that

$$(1-\varepsilon) \leqslant \left\|f(u)\right\|^2 \leqslant (1+\varepsilon), \text{ with probability } O(n^{-2}).$$

Observe that the distribution of $f(u)$ is the same (up to a multiplicative factor) of a random unit vector projected onto the first k coordinates. One can choose

$$u = \left\|X\right\|^{-1}(X_1, \ldots, X_d) \in \mathbb{R}^d \tag{2.3}$$

where $X = (X_1, \ldots, X_d)$ is a Gaussian vector with IID entries $N(0,1)$ and f the projection onto the first k coordinates, yielding

$$\left\|f(u)\right\|^2 = \left\|X\right\|^{-2} \sum_{j=1}^{k} X_j^2.$$

[1] this is the reason why there exists an equivalent formulation where $\mathbb{R}^d$ is replaced with a general Hilbert space. However the dimension d plays a role in the implementation of R (of course!).

An application of concentration inequalities shows that for $\beta < 1$ and $k < d$:

$$\mathbb{P}\left(\|f(u)\|^2 \leqslant \beta\frac{k}{d}\right) \leqslant \exp\left(\frac{k}{2}(1-\beta+\ln\beta)\right).$$

Choosing $\beta = (1-\varepsilon)$ and $Ru = \sqrt{d/k}f(u)$, the last expression becomes

$$\mathbb{P}\left(\|Ru\|^2 \leqslant 1-\varepsilon\right) \leqslant \frac{1}{n^2},$$

which translates into the fact that the probability of distorting the difference of two vectors in S by more than $1 \pm \varepsilon$ is bigger than $1 - 1/n$. We have thus proven that a random linear mapping satisfies (1.1) with high probability, this ensures the existence of a deterministic R. □

Remark 2.3. *Observe that the choice $\varepsilon = O(n^{-1/2})$ forces $k \approx n$. In applications, it is thus important to have a large number of samples, i.e., a large n, in order to choose a small ε. Differently said, there is no point in applying JL Lemma with, e.g., $n = 5000$ and $\varepsilon = 0.01$. See the dedicated page in scikit-learn documentation.*

Further improvements on the constants can be derived using refined concentration inequalities (eventually not considering Gaussian vectors in equation (2.3) but other sub-Gaussian distributions), we do not pursue this analysis in details but refer to Subsection 2.3 and to [10]. Observe that in practice it seems that the constants are not always sharp (i.e., they can be improved), see again [10].

A first implementation of R. From the proof of Theorem 2.2 we see that one can start by taking R to be a $k \times d$ matrix where the lines are orthogonal Gaussian vectors (with expected norm $\sqrt{d/k}$). Namely

(1) Construct $R \in \mathcal{M}_{k\times d}$ matrix with IID entries $\mathcal{N}(0,\sigma^2)$;
(2) Orthonormalise the rows, set $R \leftarrow (RR^\top)^{-1/2}R$.

However, the above procedure is not computationally cheap and lead to a dense matrix. We step to the next key result: quasiorthogonality!

2.2. **Quasi-orthogonality.**

Let u and v be two uniformly chosen random vectors on the unit sphere $\mathbb{S}^{k-1}$. Which is the probability that they are almost orthogonal, i.e., that their scalar product is approximately 0?

For a (finite dimensional) vector space V and $\varepsilon \in [0,1)$, we can define the ε-*quasiorthogonal dimension* of V as the largest number of points of V such that for any two of them, which are distinct, their scalar product belongs to $(-\varepsilon,\varepsilon)$. When $\varepsilon = 0$, this clearly yields the linear dimension of V.

Define the following quantity

$$\dim_\varepsilon k = \varepsilon\text{-quasiorthogonal dimension of } \mathbb{S}^{k-1}. \tag{2.4}$$

We want to understand the relationship between k, ε and $\dim_\varepsilon k$. In our case, $\dim_\varepsilon k$ is (potentially) the number of points in S, i.e., n. Indeed, we can see it as the number of orthogonal vectors that we can project onto a k-dimensional subspace, preserving their angles by a factor $(1 \pm \arccos(\varepsilon))$.

The main result which interests us, is the following one.

Theorem 2.4 ([16, Theorem 3.1 and Corollary 3.5]). *The following statements hold true:*

(1) *for every positive integer k and for every $\varepsilon \in [0, 1/\sqrt{k})$, $\dim_\varepsilon k \leqslant k\frac{1-\varepsilon^2}{1-k\varepsilon^2}$, in particular $\dim_{1/k} k = k + 1$.*

(2) *for every $\varepsilon \in (0, 1)$ and k large enough $\dim_\varepsilon k \gtrapprox 2^{k\varepsilon^2/2}$.*

Idea of the (beautiful) proof. Consider $\Gamma(k, \varepsilon)$ the (infinite) graph with vertices $\mathbb{S}^{k-1}$, where two vectors $x, y \in \mathbb{S}^{k-1}$ are connected by an edge if and only if their scalar product satisfies $|x \cdot y| \leqslant \varepsilon$. Let $G(k, \varepsilon)$ be a graph constructed in the same way but with vertices in $\{-1, 1\}^k$. Clearly $G(k, \varepsilon)$ is a finite graph and it is isomorphic to a subgraph in $\Gamma(k, \varepsilon)$ (since $\{-1/\sqrt{k}, 1\sqrt{k}\}^k \subset \mathbb{S}^{k-1}$).

Now observe that $\dim_\varepsilon k$ is equal the size of one of the largest clique[2] in $\Gamma(k, \varepsilon)$. We can thus lower-bound $\dim_\varepsilon k$ by lower-bounding the size of the largest clique in $G(k, \varepsilon)$. This can be done using a variation of Turan's Theorem[3]:

Theorem 2.5 (Berge's Theorem). *Let G be a graph with k vertices and let $\delta(G)$ represent the smallest vertex degree in G. Then each maximal clique has cardinality greater than or equal to $\lceil k/(k - \delta(G)) \rceil$.*

The statistics of $G(k, \varepsilon)$ are well-known and it is possible to explicit $\delta(G(k, \varepsilon))$ in terms of binomial coefficients (observe that $G(k, \varepsilon)$ is highly homogeneous!). This is done in [16, Lemma 3.3] and represents the last key ingredient before finishing the proof. □

Remark 2.6. *From* [16]*: it is impossible to estimate the (linear) dimension of an unknown space (but from which we can draw elements), by checking how many vectors are almost orthogonal between one another.*

From (1) we can extrapolate the same conclusion of Remark 2.3, i.e., for small ε the dimension k of the projected space has to be of order n. From (2), the fact that, for a fixed ε and a large n, k can be taken to be $O(\log n)$.

A main consequence, is the fact that the matrix R can be taken to be a Gaussian matrix with IID entries whenever d is large enough, thus removing the orthogonal requirement among the rows.

Remark 2.7. *Last proof does not say that two random vectors are nearly orthogonal, but only the fact that there are a lot of quasiorthogonal vectors*[4]*! This proof is not probabilistic and gives more insight on the structural properties of large finite-dimensional spaces. Notably, there are a lot of quasiorthogonal vertices in $\mathbb{S}^{k-1}$ because the minimum degree in $G(\varepsilon, k)$ is sufficiently large for a fixed $\varepsilon > 0$!*

2.3. **Further implementations of R.** Whenever d is large enough we can drop the orthogonal requirement on the rows of R. With high probability, the rows of R will be orthogonal, see Figure 2.3.

[2] a subset of vertices such that the induced subgraph is fully connected.

[3] `https://en.wikipedia.org/wiki/Tur%C3%A1n%27s_theorem`

[4] It is probably possible to prove JL Lemma using Theorem 2.4 and the homogeneity of $\Gamma(k, \varepsilon)$.

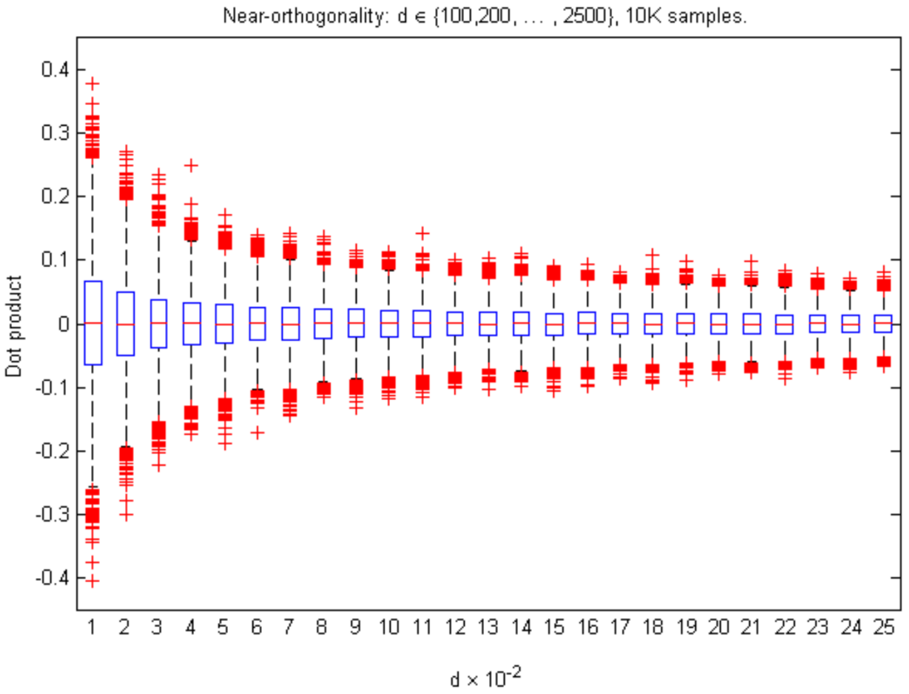

Figure 1. As d increases, the rows of R become nearly orthogonal to each other. Image taken from the (very useful) presentation [8].

Looking at the proof of Theorem 2.8, we see that any subgaussian distribution could do the job when constructing the unit vector in (2.3), since concentration inequalities apply in this case as well. This is at the core idea of the algorithm implementations by Achlioptas [1] and Ailon and Chazelle [2], known as fast JL transform.

Popular choices are given by R_{ij} Rademacher random variables as well as

$$R_{ij} = \begin{cases} +1 & \text{w.p. } q \\ -1 & \text{w.p. } q \\ 0 & \text{w.p. } 1-2q \end{cases} \qquad \text{for } q > 0. \tag{2.5}$$

We refer to [17, 19] for further improvements and the behavior of R for $q \downarrow 0$.

2.4. **Generalizations of JL Lemma.** The idea of the proof given in Theorem 2.2 can be improved to show a more general statement which implies Theorem 2.2 for a suitable choice of the parameters.

Theorem 2.8 (Randomized JL Lemma)**.** *Let $0 < \varepsilon < 1$ and an integer k such that $k \geqslant C\varepsilon^{-2} \log \delta^{-1}$, for a large enough absolute constant C. Then there is a random linear mapping $P : \mathbb{R}^d \to \mathbb{R}^k$, such that for any unit vector $x \in \mathbb{R}^d$:*

$$\mathbf{P}\left((1-\varepsilon) \leqslant \|Px\|^2 \leqslant (1+\varepsilon)\right) \geqslant 1 - \delta. \tag{2.6}$$

Proof. It is essentially a refinement of the proof given by Dasgupta and Gupta in 1999. See [8] for a detailed proof. □

An interesting extension of the randomized JL Lemma is given by the (separable) Hilbert space version.

Theorem 2.9 (JL Lemma from separable Hilbert space). *Let H be a separable Hilbert space. For $0 < \varepsilon, \delta < 1$ and any positive integer n, let k be a positive integer such that*

$$k \geqslant 4(\varepsilon^2/2 - \varepsilon^3/3)^{-1} \log \frac{n}{\sqrt{\delta}}. \tag{2.7}$$

There exists a linear map $f : H \to \mathbb{R}^k$ (which is a random projection) such that, for any set S of n points in H, with probability at least $1 - \delta$, condition (1.1) *holds for all $u, v \in S$.*

Proof. It can be found in [4, Theorem 3.1]. □

A version for finite-dimensional manifolds with bounded curvature exists as well, we refer to [3].

3. Applications: a personal selection

There are plenty of applications and the literature has become impossible to cite. We will focus on a very personal selection.

Random projection has been applied to input data (see compressed sensing and compressed learning), to output data (for classification analysis with large number of categories), to the feature space (see compressed least squares) and so on. Below we give a rather short list of references and detail a few of them in the following subsections.

A few references and keywords:

- Margin-based classifiers [20] and Linear Discriminant Analysis [9];
- Provably Learning Mixtures of Gaussians in high dimensional settings;
- Approximate kernel methods (replace kernel with a rank-1 kernel in a lower dimensional space), see [5];
- Compressed sensing (for a particular choice of the parameters, R satisfies the Restricted Isometry Property[5]) and compressed learning, see [6];
- Random projection applied to output data for classification problems with high-dimensional category space, see [15].

3.1. Approximate nearest-neighbors.

This is probably one of the first reasons to the success of JL Lemma and its use in computer science. I'm going to skip it and leave you with the classical reference [13].

3.2. Clustering in Hilbert spaces.

Step to [4].

3.3. Compressed Least Squares Regression.

Step to [18] and references therein.

[5] `https://en.wikipedia.org/wiki/Restricted_isometry_property`

3.4. **Where does random projection work poorly?** A very first numerical result is given by [11] in 2003. It is shown that PCA works better (in every experiments), but that it is significantly outperforming when applied to decision tree methods as random forest. In other words, random projection requires a large k to work efficiently with decision trees.

I've found a more rigorous motivation of the previous result in [21]. Random projection tends to uniform the input data with respect to the features. Decision trees are based on partitioning the input data with respect to the features and not necessarily with respect to the Euclidian distance.

Acknowledgments

I would like to thank Sothea Has for the two references [4, 18] and for our discussions on random projection. The presentation [8] has been extremely useful.

References

[1] D. Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687, 2003.

[2] N. Ailon and B. Chazelle. Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of Computing*, STOC '06, pages 557–563, New York, NY, USA, 2006. Association for Computing Machinery.

[3] R. G. Baraniuk and M. B. Wakin. Random Projections of Smooth Manifolds. *Foundations of Computational Mathematics*, 9(1):51–77, 2009.

[4] G. Biau, L. Devroye, and G. Lugosi. On the Performance of Clustering in Hilbert Spaces. *IEEE Transactions on Information Theory*, 54(2):781–790, 2008.

[5] A. Blum. Random Projection, Margins, Kernels, and Feature-Selection. In *Subspace, Latent Structure and Feature Selection*, Lecture Notes in Computer Science, Berlin, Heidelberg, 2006. Springer.

[6] R. Calderbank, S. Jafarpour, and R. Schapire. Compressed Learning: Universal Sparse Dimensionality Reduction and Learning in the Measurement Domain. Tech Report, Rice University, 2009.

[7] S. Dasgupta and A. Gupta. An elementary proof of the Johnson-Lindenstrauss Lemma. Tech Report, 1999.

[8] R. J. Durrant and A. Kabán. Random Projections for Machine Learning and Data Mining: Theory and Applications, 2012.

[9] R. J. Durrant and A. Kabán. Random projections as regularizers: learning a linear discriminant from fewer observations than dimensions. *Machine Learning*, 99(2):257–286, 2015.

[10] J. Fedoruk, B. Schmuland, J. Johnson, and G. Heo. Dimensionality reduction via the Johnson—Lindenstrauss Lemma: theoretical and empirical bounds on embedding dimension. *The Journal of Supercomputing*, 74(8):3933–3949, 2018.

[11] D. Fradkin and D. Madigan. Experiments with Random Projections for Machine Learning. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

[12] P. Frankl and H. Maehara. Some geometric applications of the beta distribution. *Annals of the Institute of Statistical Mathematics*, 42(3):463–474, 1990.

[13] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing - STOC '98*, pages 604–613, Dallas, Texas, United States, 1998. ACM Press.

[14] W. Johnson and J. Lindenstrauss. Extensions of Lipschitz maps into a Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

[15] A. Joly, P. Geurts, and L. Wehenkel. Random Forests with Random Projections of the Output Space for High Dimensional Multi-label Classification. In T. Calders, F. Esposito, E. Hüllermeier, and R. Meo, editors, *Machine Learning and Knowledge Discovery in Databases*, Lecture Notes in Computer Science, pages 607–622, Berlin, Heidelberg, 2014. Springer.

[16] P. C. Kainen and V. Kůrková. Quasiorthogonal dimension of euclidean spaces. *Applied Mathematics Letters*, 6(3):7–10, 1993.

[17] D. M. Kane and J. Nelson. Sparser Johnson-Lindenstrauss Transforms. *Journal of the ACM*, 61(1):4:1–4:23, 2014.

[18] O.-A. Maillard and R. Munos. Linear Regression With Random Projections. *Journal of Machine Learning Research*, (13):38, 2012.

[19] J. Matoušek. On variants of the Johnson–Lindenstrauss lemma. *Random Structures and Algorithms*, 33(2):142–156, 2008.

[20] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, pages 1177–1184, Red Hook, NY, USA, 2007. Curran Associates Inc.

[21] C. Tang and D. Garreau. When do random forests fail? In *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*, Montreal, Canada, 2008.

Email address: `fcopppini@lpsm.paris`