Very-Low Random Projection Maps

Anastasios Zouzias*
Swiss Re, Switzerland

Michail Vlachos†
IBM Research - Zurich, Switzerland

ABSTRACT

For Big Data analytics, working in low dimensionalities is beneficial for high performance. Instead of projecting onto a single low dimensionality, we examine, both analytically and empirically, the effects on the 'learning utility' of the original dataset when combining several very low-dimensional random projections. The embedding proposed exhibits many favorable traits to existing low-dimensional methodologies, such as low runtime and equivalent or better embedding quality.

1 INTRODUCTION

Random linear projections are well studied owing to the Johnson-Lindenstrauss (JL) lemma [11]. The JL lemma states that any n point-set in high-dimensional Euclidean space can be projected into $O(\log n)$ dimensions while accurately preserving all pairwise distances. The lemma is *tight*, in the sense that $\Omega(\log n)$ dimensions are necessary –see [3] and [10] for lower bounds. Although these lower bounds suggest that reducing the dimensionality of the input data further is not feasible while preserving its metric structure, this could be possible by combining the information from *several very low-dimensional* random projections. So here we examine the following: Given any Euclidean set of n points $\mathcal{P}$ in $\mathbb{R}^d$ and a target dimensionality t which is smaller than $O(\log n)$, is it possible to preserve pairwise distances by combining multiple random projections? How many independent random projections would be required? We study this particular question both from a theoretical and a practical perspective. On a theoretical aspect, we derive bounds on the expected number of random projections needed to accurately answer proximity queries (Theorem 3.1). From a practical perspective, we propose an embedding, VLR-Map, that learns the k-Neighborhood structure of the data-points.

2 RELATED WORK

Several approaches exist in the literature that use several random projections for data analysis [2, 5, 8, 12, 13]. We enumerate a representative list of these efforts. The power of several one-dimensional random projections has been exploited by Kleinberg in an algorithm for nearest-neighbor search [12]. Several one-dimensional random projections of the input data are used as proximity tests for a given query point, which is the underlying idea for constructing efficient data-structures for nearest-neighbor search. The use of multiple random projections into an arbitrary small number of dimensions for nearest neighbor search has been also studied by [2], in which the author projects randomly the original data in several independent trials and then builds a KD-tree data structure for each instance of the projected data. The set of these KD-trees are used for approximately answering nearest-neighbor queries. Designing a family of locality sensitive hashing (LSH) functions shares conceptual similarities to the framework proposed here [8, 13]. LSH principles also use random projections and aggregate the projections. Finally, data embedding techniques also follow a similar methodology. For example, Boostmap [5] learns a data embedding using triplets of inequalities; the learning process is driven by AdaBoost [7]. Boostmap works effectively in practice, but does not offer any formal analytical guarantees on the quality of its embedding, unlike the present work.

3 OUR APPROACH

The intuition behind the algorithm proposed is as follows: Although a *single* very-low-dimensional random projection might not be useful for approximately preserving all pairwise distances of a given set of points, it might be the case that *several* very-low-dimensional random projections are sufficient, when combined appropriately. There are several questions to be addressed: (1) How many dimensions should be chosen when projecting a given dataset? (2) How many different random projections are needed? (3) How should we combine these random projections? In the related literature there do not exist any satisfying answers to the first question, so far. However, we will see shortly, that as Theorem 3.1 suggests, one can set the number of dimensions t to be $\Omega(1/\varepsilon^2)$, where $\varepsilon > 0$ is the desired accuracy. Theorem 3.1 also provides a rigorous answer to the second question. Now, we turn our attention to the third question.

For the sake of presentation, assume that we want to preserve the distance between two input points $\mathbf{p}_1 \in \mathcal{P}$ and $\mathbf{p}_2 \in \mathcal{P}$ with respect to a query point $\mathbf{q} \in \mathbb{R}^d$. The proposed algorithm constructs several independent low-dimensional random projections using random matrices $G_1, G_2, \ldots, G_l$. We view each random projection as a voter that advocates on the proximity of $\mathbf{p}_1$ (or $\mathbf{p}_2$) to $\mathbf{q}$: each random projection votes for (or against) the validity of the predicate $\{\|\mathbf{p}_1 - \mathbf{q}\|_2 < \|\mathbf{p}_2 - \mathbf{q}\|_2\}$ by checking whether the corresponding *projected* points satisfy the above predicate. If $\|\mathbf{p}_1 - \mathbf{q}\|_2 \ll \|\mathbf{p}_2 - \mathbf{q}\|_2$, then it is easy to show that $\mathbf{p}_1$ will be closer to $\mathbf{q}$ than $\mathbf{p}_2$ in the projected space with at least constant probability [11]. An unbiased way to combine the votes of all random projections is to take their majority vote. More precisely, if at least half of the random projections vote that $\mathbf{p}_1$ is closer to $\mathbf{q}$ than $\mathbf{p}_2$ is, then the algorithm reports that $\mathbf{p}_1$ is closer to $\mathbf{q}$, or vice versa.

To complete the description of the above algorithm, we have to specify the required number of independent random projections. Given $\mathcal{P}$, the following theorem bounds the number of random projections that are needed for the algorithm to be effective with probability $1 - \delta$. The proof of the theorem below (given in the appendix) is based on concentration of measure arguments appropriately combined with ε-net arguments. In more detail, we build a very dense net[1] $\mathcal{N}$, i.e., $(\varepsilon/\sqrt{d})$-net, on the unit ball of

*Work conducted while at IBM Research - Zurich.

†The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement no 259569.

[1]In a metric space $M = (X, d)$ and $\varepsilon > 0$, an ε-net is a subset $\mathcal{N}$ of X so that for every $x \in X$ there exists $w \in \mathcal{N}$ so that $d(x, w) \leq \varepsilon$.

$\mathbb{R}^d$ and bound the probability that for each pair of points in $\mathcal{N}$ their norms are preserved for the majority of random projections. Although ε-net arguments of this type can be encountered previously in the relevant literature, the next theorem we state is novel, and appears, to the best of our knowledge, for the first time here.

Our main theorem states that if we draw $O(d\log(d)+\log(1/\delta))$ independent random projections then, with probability at least $1-\delta$, proximity queries between any two points of $\mathcal{P}$ can be answered correctly for well-separated points.

Theorem 3.1. *Let* $\mathrm{A} \in \mathbb{R}^{d\times n} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n]$, $0 < \delta < 1$ *and* $0 < \varepsilon < 1/2$. *Fix any integer* $t = \Omega(1/\varepsilon^2)$ *and let* $\mathrm{G}_1, \mathrm{G}_2, \ldots, \mathrm{G}_l$ *be an i.i.d. sequence of* $t \times d$ *random-sign matrices rescaled by* $1/\sqrt{t}$. *If*

$$l \geq \Omega\left(\frac{d\ln(d/\varepsilon^2)}{\varepsilon^2 t} + \ln(1/\delta)\right) \qquad (1)$$

then with probability at least $1-\delta$ *the following holds: Let* $\mathbf{p}_i, \mathbf{p}_j \in \mathrm{A}$ *and given any query* $\mathbf{q} \in \mathbb{R}^d$ *with* $(1+\gamma)\left\|\mathbf{p}_i - \mathbf{q}\right\|_2 \leq \left\|\mathbf{p}_j - \mathbf{q}\right\|_2$ *where* $\gamma > 6\varepsilon$ *then* $\left\|\mathrm{G}_k(\mathbf{p}_i - \mathbf{q})\right\|_2 < \left\|\mathrm{G}_k(\mathbf{p}_j - \mathbf{q})\right\|_2$ *holds for the majority of indices* $k \in [l]$.

Remark 1. *When we are only interested in answering nearest neighbour queries between points in* $\mathcal{P}$, *the parameter* d *in Equation* (1) *can be replaced with* n. *Indeed, repeat the proof of Theorem 3.1 by building an* ε*-net on the span of* $\mathcal{P}$.

Theorem 3.1 provides a guarantee on the preservation of nearest neighbor queries and as a direct consequence, it preserves the kNN metric structure of the input dataset. The kNN preservation property implies that the accuracy of the proposed kNN classification method converges to the accuracy of the kNN classifier in the original high dimensionality, as progressively more independent projections are used.

Discussion: Theorem 3.1 suggests that it is possible to bound the distortion even for very-low-dimensional projections. The result may appear pessimistic at first glance because it recommends a prohibitive number, for practical consideration, of $O(d\log(d))$ independent projections. It is important to consider that the analysis is necessarily pessimistic, because it is not based on the characteristics of a particular distribution or structure, but is generic. That is why the bounds may seem large. However, conditioned on the event that we possess a family of projections that satisfy the conclusion of Theorem 3.1, a Chernoff bound implies that only a constant number of projections are required.

Lemma 3.2. *Sample* S *indices from* $\{1, 2, \ldots, l\}$ *uniformly at random with replacement. If* $S \geq \frac{2(1+2\eta)^2\ln(1/\theta)}{4\eta^2}$, *then with probability at least* $1-\theta$, $0 < \theta < 1$, *the sample* S *will return the same answer as the majority over all* $\{\mathrm{G}_i\}_{i\in[l]}$.

Proof. Let $\mathcal{I}_i$ be the indicator random variable corresponding to the success of the i-th sample from S. By hypothesis, $\mathbb{E}[\mathcal{I}_i] = 1/2 + \eta$. The multiplicative Chernoff bound implies that $\Pr\left(\sum_{i=1}^{S} \mathcal{I}_i < (1-\zeta)(1/2+\eta)S\right) \leq \exp(-\zeta^2 S/2)$ for every $\zeta \geq 0$. Set $\zeta = 1 - \frac{1}{1+2\eta}$ which implies that $(1-\zeta)(1/2+\eta)S = S/2$, also $S \geq 2\ln(1/\theta)/\zeta^2$ implies that $\exp(-\zeta^2 S/2) \leq \theta$. □

Lemma 3.2 suggests that in practice there is no need to aggregate over all projections, but only over a small number of them. This is verified in the experimental section, in which we demonstrate that in practice substantially fewer number of projections are required for datasets with particular structure (i.e., real-world datasets). For all our experiments, we set an upper bound of $l = 70$ independent projections which preserved accurately the neighborhood structure. In fact, in the experimental Section 5 one can see that using multiple but lower-dimensional projections is typically better than having a single higher-dimensional projection with the same storage space. The intuition, here, is that introducing randomness is a favorable component in classification, similar, for example, to the approach that random forests also follow.

4 VLR-MAP

Using the previous theoretical results, we now present VLR-Map, standing for Very-Low Random Projection Map. It capitalizes on very-low-dimensional projections which, when combined, yield an effective embedding. The power of the embedding proposed, lies on its simplicity of implementation and low runtime cost. At its core, VLR-Map learns a low-dimensional embedding by drawing independent random projections, until the distances of the kNN neighbors over all points are sufficiently preserved through a voting process in the low-dimensional space.

Training the embedding: Assume a set of $t\times d$ random-projection matrices $\mathrm{G}_1, \mathrm{G}_2, \ldots, \mathrm{G}_l$ and an integer $1 \leq k < n$ representing the number of nearest neighbors. For each point $\mathbf{p} \in \mathcal{P}$, we define the i-th nearest neighbor of $\mathbf{p}$ (w.r.t. $\mathcal{P}$) as $\gamma_i(\mathbf{p})$ for every $1 \leq i \leq k < n$. Define the following, $\mathrm{T}(\mathbf{q}, \mathbf{p}_i, \mathbf{p}_j)$ equals 1 if $\left\|\mathbf{q} - \mathbf{p}_i\right\|_2 < \left\|\mathbf{q} - \mathbf{p}_j\right\|_2$ and -1, otherwise. Similarly, for every random projection $s = 1, 2, \ldots, l$, define

$$\widetilde{\mathrm{T}}^{(s)}(\mathbf{q}, \mathbf{p}_i, \mathbf{p}_j) := \begin{cases} 1 & \text{if } \left\|\mathrm{G}_s(\mathbf{q} - \mathbf{p}_i)\right\|_2 < \left\|\mathrm{G}_s(\mathbf{q} - \mathbf{p}_j)\right\|_2 \\ -1 & \text{otherwise} \end{cases}$$

In essence, $\widetilde{\mathrm{T}}^{(s)}(\mathbf{q}, \mathbf{p}_i, \mathbf{p}_j)$ gives us the *vote* of the projection matrix G_s regarding the proximity between the vectors $\mathbf{q}, \mathbf{p}_i$ and $\mathbf{p}_j$. Now, given several different projections/voters, one can define the majority vote over them:

$$\mathrm{Maj}(\mathbf{q}, \mathbf{p}_i, \mathbf{p}_j) := \begin{cases} \mathbf{p}_i & \text{if } \frac{1}{l}\sum_{s=1}^{l} \widetilde{\mathrm{T}}^{(s)}(\mathbf{q}, \mathbf{p}_i, \mathbf{p}_j) \geq 0 \\ \mathbf{p}_j & \text{otherwise} \end{cases}$$

Given a query $\mathbf{q}$, $\mathrm{Maj}(\mathbf{q}, \mathbf{p}_i, \mathbf{p}_j)$ reports which point between $\mathbf{p}_i$ and $\mathbf{p}_j$ is nearest to $\mathbf{q}$. Following the above discussion, we define the misclassification rate for a given set of random projections given by Eqn. (2).

$$\mathrm{Err}_k(\mathrm{A}) = \frac{1}{n\binom{k}{2}} \sum_{\mathbf{p}\in\mathcal{P}} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \mathbf{1}_{\gamma_i(\mathbf{p}) = \mathrm{Maj}(\mathbf{p}, \gamma_i(\mathbf{p}), \gamma_j(\mathbf{p}))}, \qquad (2)$$

where $\mathbf{1}_x$ is the indicator function, i.e., $\mathbf{1}_x$ equals to 1 if x is true, and zero otherwise.

The equation measures the average misclassification rate of each point $\mathbf{p} \in \mathcal{P}$ between all pairs of points in the kNN set and will be used as the measure of quality of any embedding.

Given the original high-dimensional points $\mathcal{P}$, VLR-Map learns the minimum number of random projections that are required to approximately preserve the nearest neighbors using very simple voting principles. We are only interested in preserving the nearest neighbor set of $\mathcal{P}$, so VLR-Map draws independent random projections until the misclassification rate in the kNN neighborhood is sufficiently small. We measure the error using the distance ordering over all pairs of points of the original kNN neighborhood , i.e., the ordering of the distances between any two nearest neighbor points of $\mathbf{p}$ $\gamma_i(\mathbf{p})$ and $\gamma_j(\mathbf{p})$ for $1 \leq i, j \leq k$. VLR-Map is scalable because its complexity is essentially linear to the dataset size; the algorithm has an $O(nk^2)$ cost per iteration and the bound of Theorem 3.1 points to the fact that the algorithm will terminate after a finite number of iterations. In the experimental section, we provide further empirical validation on the quality and runtime of our technique and compare it with other embedding methodologies.

Answering kNN queries: After execution of VLR-Map, the resulting set of projected instances of $\mathcal{P}$ can be used for answering proximity queries. Given a query point $\mathbf{q} \in \mathbb{R}^d$ and any two points $\mathbf{p}_i, \mathbf{p}_j \in \mathcal{P}$ we can test whether $\mathbf{q}$ is closer to $\mathbf{p}_1$ or $\mathbf{p}_2$ by using $\text{Maj}(\mathbf{q}, \mathbf{p}_i, \mathbf{p}_j)$. A crucial computational feature when evaluating $\text{Maj}(\cdot, \cdot, \cdot)$ is that it can be effectively approximated by random sampling as previously explained in Lemma 3.2. In practice, a constant number of projections are sufficient for approximately answering proximity queries. So, for any any unlabeled query point $\mathbf{q}$, the algorithm will provide a label based on the consensus voting from all the very-low dimensional classifiers.

5 EXPERIMENTS

We examine the performance and quality of the algorithms presented on several publicly available datasets. [4, 15]. All the datasets are high-dimensional (with dimensionalities varying from 100+ to 10,000), and while they are not very big datasets, they serve well for showcasing the differences in performance and accuracy of the techniques compared. We compare the embedding quality of our approach with traditional random-projection approaches which project on a higher dimensionality that uses the same total space as our methodology. We show that our approach exhibits better classification accuracy and briefly analyze this result.

5.1 Validation of Main Theorem

First, we provide empirical validation for our main result of Theorem 3.1. Recall that the Theorem states essentially that as we increase the number of independent projections, preservation of nearest neighbor structure will be progressively better. Figure 1 plots the embedding error when preserving the 3-NN structure as we progressively increase the number of independent projections. For clarity, we plot the results on four datasets: `Email`, `Gisette`, `USPS` and `MUSK`. The results for the other datasets exhibit similar pattern and are omitted.

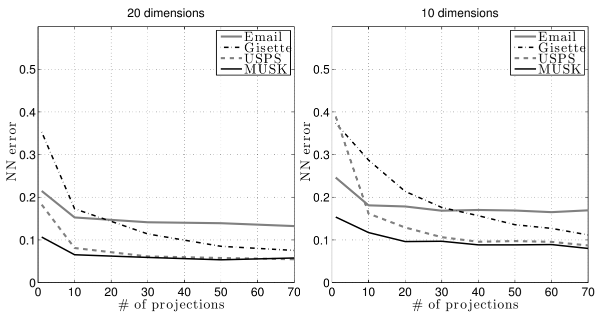

Figure 1: Empirical validation of Theorem 3.1. When projecting to 10 dimensions (right) instead of 20 (left) an equivalent NN-error can be achieved by using additional independent projections.

We use two target dimensions, $t = 20$ and $t = 10$ and average the results over ten independent executions. Observe that Figure 1 validates the main result, because by using a lower target dimensionality of $t = 10$ an equivalent error of the higher target dimensionality $t = 20$ can still be achieved through the use of additional projections.

Therefore, the power of the methodology proposed lies in its simplicity; by using and combining additional very-low-dimensional projections we can substantially influence the quality of the embedding and of the distance preservation. It is important to underscore that because the individual projections are independent of each other, the classifiers operating on each of the projections are totally segregated and could be run in parallel. Because each classification is independent of the others (aside from the small voting phase), given sufficient CPUs/cores, the overall runtime of our approach should remain approximately constant, even under increasing cardinality of projections/classifiers.

5.2 Random Projection Methodologies

We compare the classification error of various methodologies based on random projections: VLR-Map, a kNN classifier using a *single* projection onto t dimensions (sRP_t), a kNN classifier with a *single* projection onto $t \cdot l$ dimensions ($\mathsf{sRP}_{l \cdot t}$) and locality-sensitive-hashing (LSH) [8]. $\mathsf{sRP}_{l \cdot t}$ is included in the comparisons to compare the performance of VLR-Map with the traditional single random projection methodology which *uses the same space* ($\mathsf{sRP}_{l \cdot t}$ has the same number of coordinates with our approach). The comparison with LSH is also done under fair settings; the number of hash functions equals the number of projections l of VLR-Map, and the number of bins for each hash function equals the projected dimensionality t. Finally, for reference, we also include the classification accuracy of the kNN classifier on the original high-dimensional points (`kNN`). We report the results for $k = 3$ nearest neighbors and target dimensionality of $t = 30$ in Table 1. The experiments indicate that our approach can, in fact, achieve comparable (or sometimes even better) performance than the kNN classifier which operates in the original data dimensionality. An advantage of our framework is that it is computationally lighter than a traditional kNN classifier, because in very low dimensional spaces (in which our framework operates) the nearest-neighbor search can be executed efficiently using data structures, such as KD-trees. However, in higher-dimensional spaces, the performance of these techniques degrades rapidly as validated both analytically and empirically in many studies [9].

More importantly, the results suggest that our approach outperforms the traditional projection methodology which uses the same space (i.e., a single random projection at dimensionality $t \cdot l$). One can think of this as quite analogous to the concept behind random forests [6]. Having multiple random classifiers (unweighted in our case) can boost classification and also introduce robustness. Finally, in Table 2 we report the runtime for one experiment on three datasets for the various techniques based on random projections. VLR-Map and $\mathsf{sRP}_{t \cdot l}$ have equivalent runtime, while LSH is costlier.

6 CONCLUSION

Our main theorem highlights that it is feasible to combine many very-low-dimensional projections and guarantee a bounded distortion on the original distances. From a practical viewpoint, the embedding proposed, VLR-Map, exhibits many favorable traits, such as: i) simplicity of implementation, and, ii) scalability: significantly reduced run-time compared to state-of-art embedding techniques with comparable accuracy.

Classification Error on test data (%)														
Method	3-NN	sRP$_t$	VLR-Map	sRP$_{t\cdot l}$	LSH	VLR-Map	sRP$_{t\cdot l}$	LSH	VLR-Map	sRP$_{t\cdot l}$	LSH	VLR-Map	sRP$_{t\cdot l}$	LSH
no. projections l =			**10**			**30**			**50**			**70**		
MNIST	2.95	9.25	4.29	3.23	10.14	3.73	3.11	3.77	3.61	3.12	4.64	3.5	3.15	8.95
COIL	1.15	2.3	0.92	1.23	0.69	0.85	1.15	1.85	0.38	1.15	5.0	0.77	1.23	11.47
Email	18	19.84	13.71	15.76	26.73	12.91	16.04	26.78	12.38	16.38	26.69	12.27	16.6	26.69
lights	1.9	6 .06	2.41	2.49	3.96	2.02	2.33	3.96	2.33	2.25	3.96	2.3	2.33	3.81
PIE	11.7	22.5	12.2	12.02	16.49	10.1	11.51	18.36	9.8	11.98	19.76	9.49	11.79	22.21
USPS	5.7	11.5	6.53	5.7	6.77	5.23	5.47	12.5	4.97	5.53	45.33	4.67	5.87	71.53
GISETTE	3.0	25.1	13.17	6.07	2.6	7.8	3.9	2.6	6.7	3.6	2.6	6.07	3.3	2.6
MUSK	4.21	9.33	4.98	5.82	9.47	4.91	5.89	10.25	4.98	5.47	12.00	4.42	5.12	16.07

Table 1: VLR-Map offers overall better accuracy (on t dimensions using l projections) than LSH or a single projection that uses that same space (sRP$_{t\cdot l}$). Red color denotes better values (smaller classification error).

	sRP$_{t\cdot l}$	VLR-Map	LSH
MNIST	0.87	1.22	1.95
USPS	0.06	0.10	0.85
MUSK	0.01	0.04	0.34

Table 2: Time comparison (in sec) between a single random projection (sRP$_{t\cdot l}$), VLR-Map (l = 30 and t = 10) and LSH. VLR-Map and single random projection exhibit equivalent runtime, while LSH computations are more expensive.

REFERENCES

[1] D. Achlioptas. Database-friendly Random Projections: Johnson-Lindenstrauss with Binary Coins. *J. Comput. Syst. Sci.*, 66(4):671–687, 2003.
[2] Y. S. Ahmed. Multiple Random Projections for Fast, Approximate Nearest Neighbor Search in High Dimensions.
[3] N. Alon. Problems and results in extremal combinatorics, part i. *Discrete Math*, 273, 2003.
[4] A. Asuncion and D. J. Newman. UCI machine learning repository, 2007.
[5] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios. Boostmap: An embedding method for efficient nearest neighbor retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30:89–104, 2008.
[6] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
[7] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, Aug. 1997.
[8] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. *In Proc. VLDB*, pages 518–529, 1999.
[9] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proc. STOC*, pages 604–613, 1998.
[10] T. S. Jayram and D. P. Woodruff. Optimal Bounds for Johnson-Lindenstrauss Transforms and Streaming Problems with Sub-Constant Error. In *Proc. SODA*, pages 1–10, 2011.
[11] W. B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary mathematics*, 26(189-206):1–1, 1984.
[12] J. M. Kleinberg. Two Algorithms for Nearest-Neighbor Search in High Dimensions. In *Proc. STOC*, pages 599–608. ACM, 1997.
[13] E. Kushilevitz, R. Ostrovsky, and Y. Rabani. Efficient Search for Approximate Nearest Neighbor in High Dimensional Spaces. In *Proc. STOC*, pages 614–623, 1998.
[14] J. Matousek. *Lectures on Discrete Geometry.* Springer, 2002.
[15] K. P. Murphy. A collection of MATLAB data sets used by PMTK. https://github.com/probml/pmtkdata, 2014.

Proof. of Theorem 3.1: Assume two points $\mathbf{x}, \mathbf{y}$ of the pointset A and a query point $\mathbf{q} \in \mathbb{R}^d$. The goal is to decide whether $\mathbf{q}$ is nearest to $\mathbf{x}$ or to $\mathbf{y}$. Without loss of generality, we can assume that $\mathbf{q}$ is the origin by linearity (otherwise apply the argument below to the vectors $\mathbf{x} - \mathbf{q}$ and $\mathbf{y} - \mathbf{q}$). So, it suffices to argue that one can check the distances of $\mathbf{x}$ and $\mathbf{y}$.

Let G be a $t \times d$ random-sign matrix rescaled by $1/\sqrt{t}$, i.e., a matrix whose entries are i.i.d. uniformly distributed r.v. on $\{\pm 1\}$. For any $0 < \varepsilon < 1/2$ and $\mathbf{z} \in \mathbb{R}^d$, the following bound is well-known [1]

$$\mathbb{P}\left(\left|\frac{\|\mathrm{G}\mathbf{z}\|_2^2}{\|\mathbf{z}\|_2^2} - 1\right| > \varepsilon\right) \leq \exp(-\mathrm{C_{JL}}\varepsilon^2 t) \tag{3}$$

where $\mathrm{C_{JL}} > 0$ is an absolute constant. More precisely, $\mathrm{C_{JL}}$ is the constant of the Johnson-Lindenstrauss lemma with random sign matrices. For any $\mathbf{z}_1, \mathbf{z}_2 \in \mathcal{S}^{d-1}$, define the following event

$$\mathcal{E}_j(\mathbf{z}_1, \mathbf{z}_2) := \left\{\left|\left\|\mathrm{G}_j\mathbf{z}_1\right\|_2^2 - 1\right| < \varepsilon \text{ and } \left|\left\|\mathrm{G}_j\mathbf{z}_2\right\|_2^2 - 1\right| < \varepsilon\right\}.$$

It follows by (3) that for every $j = 1, \dots, l$: $\mathbb{P}\left(\mathcal{E}_j^c(\mathbf{z}_1, \mathbf{z}_2)\right) \leq 2\exp(-\mathrm{C_{JL}}\varepsilon^2 t)$. Moreover, let us define the event $\mathrm{Maj}(\mathbf{z}_1, \mathbf{z}_2) = \{\mathcal{E}_j(\mathbf{z}_1, \mathbf{z}_2)$ holds for $\geq \lceil l/2 \rceil$ indices $j.\}$. Let us first bound the probability

$$\begin{aligned}\mathbb{P}\left(\mathrm{Maj}(\mathbf{z}_1, \mathbf{z}_2)^c\right) &= \sum_{s=\lceil l/2\rceil}^{l} \mathbb{P}\left(\mathcal{E}_j(\mathbf{z}_1, \mathbf{z}_2)^c \text{ for exactly } s \text{ indices}\right)\\ &\leq \sum_{s=\lceil l/2\rceil}^{l} \binom{l}{s}\mathbb{P}\left(\mathcal{E}_1(\mathbf{z}_1, \mathbf{z}_2)^c\right)^s \leq \sum_{s=\lceil l/2\rceil}^{l} \binom{l}{s} 2^s \exp(-\mathrm{C_{JL}}\varepsilon^2 st)\\ &\leq \sum_{s=\lceil l/2\rceil}^{l} \binom{l}{s} 2^l \exp(-\mathrm{C_{JL}}\varepsilon^2 \lceil l/2\rceil t)\\ &\leq 2^l \exp(-\mathrm{C_{JL}}\varepsilon^2 \lceil l/2\rceil t) \sum_{s=0}^{l}\binom{l}{s} \leq 2^{2l}\exp(-\mathrm{C_{JL}}\varepsilon^2 tl/2).\end{aligned}$$

Now, we bound the probability that the majority of random projections $\{\mathrm{G}_j\}$ preserves the norms of every pair of points in $\mathcal{N}$, where $\mathcal{N}$ is an $(\varepsilon/\sqrt{d})$-net of $\mathcal{S}^{d-1}$. Recall that $|\mathcal{N}| \leq (3\sqrt{d}/\varepsilon)^d$ [14]. Namely, we bound the following $\mathbb{P}\left(\forall \mathbf{z}_1 \in \mathcal{N}, \forall \mathbf{z}_2 \in \mathcal{N} : \mathrm{Maj}(\mathbf{z}_1, \mathbf{z}_2)\right) = 1 - \mathbb{P}\left(\exists \mathbf{z}_1, \mathbf{z}_2 \in \mathcal{N}, \mathrm{Maj}(\mathbf{z}_1, \mathbf{z}_2)^c\right)$. The last quantity can be bounded as follows:

$$\begin{aligned}&\mathbb{P}\left(\exists \mathbf{z}_1, \mathbf{z}_2 \in \mathcal{N}, \mathrm{Maj}(\mathbf{z}_1, \mathbf{z}_2)^c\right) \leq \sum_{\mathbf{z}_1, \mathbf{z}_2 \in \mathcal{N}} \mathbb{P}\left(\mathrm{Maj}(\mathbf{z}_1, \mathbf{z}_2)^c\right)\\ &\leq |\mathcal{N}|^2 2^{2l}\exp(-\mathrm{C_{JL}}\varepsilon^2 tl/2) \leq \left(\frac{\sqrt{d}}{3\varepsilon}\right)^{2d} 2^{2l}\exp(-\mathrm{C_{JL}}\varepsilon^2 tl/2)\\ &= \exp(-l(\mathrm{C_{JL}}\varepsilon^2 t/2 - \ln(4)) + 3d\ln(d/\varepsilon^2)),\end{aligned}$$

where in the first inequality we used the union bound and the final inequality by (3). Assume that $\varepsilon^2 t > 2\ln(4)/\mathrm{C_{JL}}$ ($t = \Omega(1/\varepsilon^2)$ by assumption), hence if $l \geq \frac{4d\ln(d/\varepsilon^2)}{\mathrm{C_{JL}}\varepsilon^2 t - 2\ln(4)} + \ln(1/\delta)$ then

$$\mathbb{P}\left(\forall \mathbf{z}_1 \in \mathcal{N}, \forall \mathbf{z}_2 \in \mathcal{N} : \mathrm{Maj}(\mathbf{z}_1, \mathbf{z}_2)\right) \geq 1 - \delta.$$

From now on, assume that the following event holds

$$\left\{\forall \mathbf{z}_1 \in \mathcal{N}, \forall \mathbf{z}_2 \in \mathcal{N} : \mathrm{Maj}(\mathbf{z}_1, \mathbf{z}_2)^c\right\}. \tag{4}$$

Next we prove that (assuming (4)) if $(1+\gamma)\|\mathbf{x}\|_2 \leq \|\mathbf{y}\|_2$, then the majority of the fixed projections $\{\mathrm{G}_i\}_{i\in[l]}$ will satisfy $\|\mathrm{G}_i\mathbf{x}\|_2 < \|\mathrm{G}_i\mathbf{y}\|_2$.

Indeed, let $\mathbf{x}_N$ and $\mathbf{y}_N$ be the net points that are nearest to $\mathbf{x}/\|\mathbf{x}\|_2$ and $\mathbf{y}/\|\mathbf{y}\|_2$, respectively. Namely, it holds that $\|\mathbf{x}/\|\mathbf{x}\|_2 - \mathbf{x}_N\|_2 \leq \varepsilon/\sqrt{d}$ and $\|\mathbf{y}/\|\mathbf{y}\|_2 - \mathbf{y}_N\|_2 \leq \varepsilon/\sqrt{d}$. Conditioning on the event in Eq. (4) implies that for the majority of $\{\mathrm{G}_j\}_{j\in[l]}$

$$\begin{aligned}\left\|\mathrm{G}_j\mathbf{x}\right\|_2 &= \|\mathbf{x}\|_2 \left\|\mathrm{G}_j\mathbf{x}/\|\mathbf{x}\|_2 - \mathrm{G}_j\mathbf{x}_N + \mathrm{G}_j\mathbf{x}_N\right\|_2\\ &\leq \|\mathbf{x}\|_2 \left(\left\|\mathrm{G}_j\right\|_2 \|\mathbf{x}/\|\mathbf{x}\|_2 - \mathbf{x}_N\|_2 + \left\|\mathrm{G}_j\mathbf{x}_N\right\|_2\right)\\ &\leq \|\mathbf{x}\|_2 \left(\left\|\mathrm{G}_j\right\|_2 \varepsilon/\sqrt{d} + 1 + \varepsilon\right) \leq (1+2\varepsilon)\|\mathbf{x}\|_2,\end{aligned}$$

where the first inequality is triangle inequality combined with standard matrix norms, the second inequality follows by Eq. (4) and the definition of $\mathbf{x}_N$ and the last inequality follows since $\left\|\mathrm{G}_j\right\|_2 \leq \left\|\mathrm{G}_j\right\|_F \leq \sqrt{d}$.

Similarly, for the majority of the indices j,

$$\begin{aligned}\left\|\mathrm{G}_j\mathbf{y}\right\|_2 &= \|\mathbf{y}\|_2 \left\|\mathrm{G}_j\mathbf{y}/\|\mathbf{y}\|_2 - \mathrm{G}_j\mathbf{y}_N + \mathrm{G}_j\mathbf{y}_N\right\|_2\\ &\geq \|\mathbf{y}\|_2 \left(\left\|\mathrm{G}_j\mathbf{y}_N\right\|_2 - \left\|\mathrm{G}_j\right\|_2 \|\mathbf{y}/\|\mathbf{y}\|_2 - \mathbf{y}_N\|_2\right)\\ &\geq \|\mathbf{y}\|_2 \left(1 - \varepsilon - \left\|\mathrm{G}_j\right\|_2 \varepsilon/\sqrt{d}\right) \geq (1-2\varepsilon)\|\mathbf{y}\|_2.\end{aligned}$$

Therefore, we conclude that the ratio $\left\|\mathrm{G}_j\mathbf{y}\right\|_2 / \left\|\mathrm{G}_j\mathbf{x}\right\|_2$ is at least $\frac{(1-2\varepsilon)\|\mathbf{y}\|_2}{(1+2\varepsilon)\|\mathbf{x}\|_2} \geq \frac{(1-2\varepsilon)}{(1+2\varepsilon)}(1+\gamma) > 1$ for the majority of random projections $\{\mathrm{G}_j\}_{j\in[l]}$ as $\gamma > 6\varepsilon$. ☐