Accepted in: Proceedings of Discrete Mathematics Days 2022, Santander.

Random projections for the distance geometry problem

Leo Liberti*[1], Benedetto Manca†[2], and Pierre-Louis Poirion‡[3]

[1]LIX CNRS, École Polytechnique, Institut Polytechnique de Paris, 91128 Palaiseau, France
[2]Dip. Matematica e Informatica, Università degli Studi di Cagliari, Via Ospedale 72, 09124, Italy
[3]RIKEN Center for Advanced Intelligence Project, Tokyo, Japan

Abstract

Random projections decrease the dimensionality of a finite set of vectors while ensuring approximate congruence, up to a multiplicative constant. Based on the theory of random projections in conic programming we derive an application of random projections to a nonconvex mathematical programming problem in distance geometry, namely that of finding the positions of the vertices of a graph in a vector space of given dimension, while ensuring that every pair of adjacent vertices is placed at a Euclidean distance equal to the corresponding edge weight.

1 Introduction

This paper is about the application of Random Projections (RP) to the Distance Geometry Problem (DGP). Insofar as RPs have been applied to Mathematical Programming (MP), this is the first time that RPs are successfully applied to a problem with nonconvex constraints.

The DGP is the following problem [9]. Given an integer $K > 0$ and a simple edge-weighted undirected graph $G = (V, E, d)$ with $d : E \to \mathbb{R}_+$, determine if there is a realization $x : V \to \mathbb{R}^K$ such that

$$\forall i < j \in V \quad \|x_i - x_j\|_2^2 = d_{ij}^2. \tag{1}$$

The DGP is useful to model inverse problems where the input is given by a subset of Euclidean distances, and the problem is to construct a consistent realization. Although the DGP is framed as a decision problem, it is defined over the real numbers, which means that it is unlikely to be in **NP** for all $K > 1$. Nonetheless, it is **NP**-hard [15].

Given $A = \{A_1, \ldots, A_n\} \in \mathbb{R}^m$, $\epsilon \in (0, 1)$, and $k = O(\epsilon^{-2} \ln(n))$, RPs are σ-sparse $k \times m$ random matrices, sampled componentwise[1] from $\mathsf{N}(0, 1/\sqrt{k\sigma})$, such that

$$\mathbf{Prob}\big(\forall i < j \leq n \quad (1-\epsilon)\|x_i - x_j\|_2 \leq \|Tx_i - Tx_j\|_2 \leq (1+\epsilon)\|x_i - x_j\|_2 \ \big) \geq 1 - \delta, \tag{2}$$

where $\delta = O(e^{-\mathcal{C}\phi(k)})$, $\mathcal{C}$ is a universal constant not depending on input data, and ϕ is usually linear in k. Eq. (2) is known as the Johnson-Lindenstrauss Lemma (JLL) [6], the main result in this area. It proves that RPs guarantee approximate congruence, with arbitrarily high probability (wahp) on finite point sets reduced from m to k dimensions. We note that the JLL is a high-dimensional theoretical result: if m is too small, any decent value for ϵ will already set k larger than m, trivializing the result.

We note outright that the JLL cannot be trivially applied to the dimensionality K of the solution vectors involved in Eq. (1). Not only is K usually too small for this purpose, but the x symbols in

*Email: liberti@lix.polytechnique.fr.
†Email: bmanca@unica.it. Partly supported by grant STAGE, Fondazione Sardegna 2018.
‡Email: pierre-louis.poirion@riken.jp.

[1]Other distributions can be used, some of which discrete, e.g. [1], but in our empirical experience in projecting continuous problems what really counts is the sparsity structure of T, rather than the values of the nonzeros.

Eq. (1) are decision variables, rather than given vectors. Given that they range over $\mathbb{R}^K$, they represent uncountable point sets: the JLL does not apply to such cases.

The key to a successful application of RPs to the DGP goes through a theory of RPs in conic programming developed in [10].

1.1 Some results about the DGP

The DGP has applications in many different dimensions. Time-synchronization network protocols motivate the DGP in one dimension $K = 1$ [16], localization of sensor networks motivate it for $K = 2$ [4], protein conformation by Nuclear Magnetic Resonance (NMR) [12] motivates it for $K = 3$. Graph embeddings in machine learning motivate it for high-dimensional values of K too [7]. For certain graph structures, one may use specific mixed-combinatorial methods [9]. In general, though, we resort to MP, a formal language for describing and solving optimization problems [8].

In this paper we make use of two simple MP formulations of the DGP: the *slack formulation*

$$\mathsf{P} \equiv \begin{cases} \min\limits_{\substack{x\in\mathbb{R}^{nK} \\ s^+\geq 0, s^-\geq 0}} & \sum\limits_{\{i,j\}\in E} (s^+_{ij} + s^-_{ij}) \\ \forall \{i,j\} \in E & \|x_i - x_j\|_2^2 \;=\; d_{ij}^2 + s^+_{ij} - s^-_{ij}, \end{cases} \tag{3}$$

and the *quartic formulation*

$$\min_{x\in\mathbb{R}^{nK}} \sum_{\{i,j\}\in E} (\|x_i - x_j\|_2^2 - d_{ij}^2)^2. \tag{4}$$

We note that both are non-convex Nonlinear Programs (NLP), and that both are exact reformulations of one another.

1.2 Some results about RPs

There is by now an extensive body of work on RPs [5, 17], ranging from new JLL proofs, to efficient ways to sample RPs, to applications in data science, to computational testing. More recently, the field of application of RPs was generalized to optimization problems, which poses new challenges: (a) dealing with the fact that decision variables may represent infinite (rather than finite) vector sets, and (b) inferring approximate feasibility and optimality from approximate congruence. Least-squares optimization subject to black-box convex constraints was discussed in [14], Linear Programming (LP) in [19], Conic Programming (CP) in [10], Quadratic Programming (QP) in [3].

The work most relevant to the current paper is [10], which gives derived approximate feasibility and optimality results for LP, Second-Order Cone Programming (SOCP) and Semidefinite Programming (SDP) using the language of Formally Real Jordan Algebras.

2 Applying RPs to the slack formulation

We let P be the DGP formulation in Eq. (3), and T be a random projector. We define a *projected DGP formulation* $T\mathsf{P}$ similarly to [10]. For a standard-form CP

$$\mathsf{C} \equiv \min\{\langle C, X\rangle \mid \forall i \leq m \; \langle A_i, X\rangle = b_i \wedge X \succeq 0\}$$

we write the linear constraints as $[A \odot X = b] \equiv [\forall i \leq m \; \mathsf{vec}(A)_i{}^\top \mathsf{vec}(X) = b_i]$, where $\mathsf{vec}(M)$ is the vector obtained by stacking the columns of M, and define the projected CP

$$T\mathsf{C} \equiv \min\{\langle C, X\rangle \mid TA \odot X = Tb \wedge X \succeq 0\}.$$

The result of this projection is that the number of constraints goes from m down to k (see Eq. (2)), which hopefully implies that it can be solved more efficiently.

First, we note that we can limit the scope of RPs to a given subset of rows. If our linear system $A \odot X = b$ consists of two stacked subsystems $A^1 \odot X = b^1$ and $A^2 \odot X = b^2$, it suffices to define $\hat{T} = \begin{pmatrix} I & 0 \\ 0 & T \end{pmatrix}$ to obtain the projected system $\hat{T}A \odot X = \hat{T}b$ consisting of the two stacked subsystems $A^1 \odot X = b^1$ and $TA^2 \odot X = Tb^2$. This allows us to project subsets of rows of a MP formulation, and still claim validity of all of the results in [10].

Next, we note that Eq. (1) can be reformulated exactly as follows:

$$\left.\begin{array}{rcl} \forall \{i,j\} \in E \quad X_{ii} - 2X_{ij} + X_{jj} & = & d_{ij}^2 \\ X & = & xx^\top. \end{array}\right\} \tag{5}$$

This reformulation is at the basis of the SDP relaxation of the DGP, which replaces $X = xx^\top$ by $X \succeq 0$. The linear subsystem $A \odot X = b$, written as $\forall \{i,j\} \in E \ \langle A^{ij}, X\rangle = d_{ij}^2$, is defined on a set of $m = |E|$ symmetric $n \times n$ matrices A^{ij} having $\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$ as their $\{i,j\}$-th main 2×2 minor.

We now apply a $k \times n^2$ RP T to Eq. (5) so that only the first m rows are projected. For the $\{i,j\}$-th row of $A \odot X = b$, the application of T to $(\mathsf{vec}(A^{ij}), d_{ij}^2)$ yields a random combination of all of the rows of (A, b), hence:

$$\forall h \leq k \quad \mathsf{vec}((TA)_h) = \sum_{\{i,j\} \in E} T_{h,\{i,j\}} \mathsf{vec}(A^{ij}) = \sum_{\{i,j\} \in E} T_{h,\{i,j\}} d_{ij}^2.$$

In particular, we have

$$\forall h \leq k \quad \sum_{\{i,j\} \in E} T_{h,\{i,j\}} (X_{ii} - 2X_{ij} + X_{jj} - d_{ij}^2) = 0. \tag{6}$$

We now substitute all of the X variables back with their definition $X = xx^\top$, which yields:

$$\forall h \leq k \quad \sum_{\{i,j\} \in E} T_{h,\{i,j\}} (\|x_i - x_j\|_2^2 - d_{ij}^2) = 0. \tag{7}$$

The derivation of Eq. (7) is valid as long as the DGP instance is feasible (otherwise [10, Thm. 3.2] no longer holds, since it is based on conic duality). We address this issue heuristically, by employing Eq. (7) in the slack DGP formulation, which is always feasible. Our projected DGP formulation turns out to be:

$$T\mathsf{P} \equiv \left\{ \begin{array}{ll} \min\limits_{\substack{x \in \mathbf{R}^{nK} \\ s^+ \geq 0, s^- \geq 0}} & \sum\limits_{h \leq k} (s_h^+ + s_h^-) \\ \forall h \leq k & \sum\limits_{\{i,j\} \in E} T_{h,\{i,j\}} (\|x_i - x_j\|_2^2 - d_{ij}^2) \ = \ s_h^+ - s_h^-. \end{array} \right. \tag{8}$$

As for the error of Eq. (7) w.r.t. Eq. (5), we prove the following result.

Theorem 1. *Assume the DGP instance $G = (V, E, d)$ is feasible, and $\bar{X}$ is a solution of Eq. (6) that also satisfies $\mathsf{rk}(\bar{X}) = \bar{x}\bar{x}^\top$ for some $\bar{x} \in \mathbb{R}^{nK}$. Then*

$$\forall u > 0 \quad \mathsf{Prob}\big[\|A \odot \bar{X} - b\|_2 \leq \delta\theta^2(\mathcal{C}\sqrt{\ln n} + 2u)/\sqrt{k}\big] \geq 1 - 2e^{-u^2},$$

where $\delta = \max\limits_{v \in V} \sqrt{\deg(v)}$, θ is an upper bound to $\|x\|_1$ over every feasible realization x of the DGP, and $\mathcal{C}$ is a universal constant.

Proof. Let $C = \{A \odot X - b \mid X = xx^\top \wedge \|x\|_1 \leq \theta\}$. By [18, Ex. 9.1.8], we have that, for all $u > 0$, the following holds with probability $1 - 2e^{-u^2}$:

$$\sup_{A \odot X - b \in C} \big| \, \|TA \odot X - Tb\|_2 - \sqrt{k}\|A \odot X - b\|_2 \, \big| \leq \mathcal{C}w(C) + u\mathsf{rad}(C),$$

where $w(C)$ is the Gaussian width and $\mathsf{rad}(C) = \sup_{Y\in C}\|Y\|_2$. Since $A\bar{X} - b \in C$, we can apply the above to $\bar{x}$. Since $TA\bar{X} = Tb$ by definition of $\bar{X}$, we obtain: $\sqrt{k}\|A\bar{X} - b\|_2 \leq \mathcal{C}w(C) + u\mathsf{rad}(C)$.

We now compute estimates of $w(C)$ and $\mathsf{rad}(C)$. By the definition of the Gaussian width, we have $w(C) = \mathbb{E}_{g\sim\mathsf{N}(0,I_m)}\sup_{A\bar{X}-b\in C}\langle g, A\bar{X} - b\rangle$. Since Gaussian widths are invariant by affine translations, we obtain $w(C) = \mathbb{E}_{g\sim\mathsf{N}(0,I_m)}\sup_{A\bar{X}\in C+b}\langle A^\top g, \mathsf{vec}(X)\rangle$. By definition of C, we replace X by $xx^\top$ and rewrite the sup quantification, yielding $w(C) = \mathbb{E}_{g\sim\mathsf{N}(0,I_m)}\sup_{\|x\|_1\leq\theta}\langle A^\top g, \mathsf{vec}(xx^\top)\rangle$. By Hölder's inequality, $\langle A^\top g, \mathsf{vec}(xx^\top)\rangle \leq \|A^\top g\|_\infty \|\mathsf{vec}(xx^\top)\|_1$. Now, note that

$$\|\mathsf{vec}(xx^\top)\|_1 = \sum_{ij}\Big|\sum_h x_{ih}x_{jh}\Big| \leq \sum_{ijh}|x_{ih}|\,|x_{jh}| \leq \theta^2 \tag{9}$$

by the triangular inequality, so $w(C) = \theta^2\mathbb{E}_{g\sim\mathsf{N}(0,I_m)}\|A^\top g\|_\infty$. Next, note that the p-th component of $A^\top g$ is distributed like $\mathsf{N}(0, \|A^p\|_2)$, so by [18, Ex. 2.5.10], we obtain $w(C) = \theta^2\mathcal{C}\max_{p\leq n^2}\|A^p\|_2\sqrt{\ln n^2}$. By re-defining $\mathcal{C}$ as $\mathcal{C}\sqrt{2}$ we get $w(C) = \theta^2\mathcal{C}\max_{p\leq n^2}\|A^p\|_2\sqrt{\ln n}$. Note that, by the structure of A in the DGP, for the p-th vertex couple $(v, z) \in V \times V$, the p-th column A^p of A has at most a single -1 entry if $v \neq z$, and $\deg(v)$ 1 entries if $v = z$, so $\max_p\|A^p\|_2 = \max_{v\in V}\sqrt{\deg(v)}$.

Since the DGP instance is feasible, there exists $\hat{X}$ s.t. $A\hat{X} = b$ and $\hat{X} = \hat{x}\hat{x}^\top$. We therefore have $\mathsf{rad}(C) = \sup_{A\bar{X}-b\in C}\|A(\bar{X} - \hat{X})\|_2$ (by replacement of b with $A\hat{X}$), whence

$$\mathsf{rad}(C) \leq \sup_{\|x\|_1\leq\theta}\|A\,\mathsf{vec}(xx^\top)\|_2 + \|A\,\mathsf{vec}(\hat{x}\hat{x}^\top)\|_2$$

by the triangular inequalities and $X = xx^\top$. Now we have

$$\|A\,\mathsf{vec}(xx^\top)\|_2 \leq \sum_{p\leq n^2}\|A^p\|_2\Big|\sum_h x_{i_ph}x_{j_ph}\Big| \leq \sup_p\|A^p\|_2\,\theta^2 = \theta^2\max_{v\in V}\sqrt{\deg(v)}$$

by Eq. (9), where (i_p, j_p) is the p-th vertex pair in $V \times V$. □

We also note that θ can be computed from the DGP instance $G = (V, E, d)$ by assuming that the realization centroid is at the origin, and taking the worst case of a realization x on a single segment: the realization diameter is then $\sum_{\{i,j\}\in E} d_{ij}$, whence one can easily derive bounds for x and hence θ.

2.1 A new DGP solution algorithm

We note that Eq. (8) is another nonconvex NLP, albeit with fewer constraints than Eq. (3). We can therefore only solve it locally in practically acceptable times.

Instead, we turn to Barvinok's naive algorithm applied to the DGP [11]: the solution $\bar{X}$ of the SDP relaxation of Eq. (5)

$$\left.\begin{array}{rrcl} \forall\{i,j\}\in E & X_{ii} - 2X_{ij} + X_{jj} & = & d_{ij}^2 \\ & X & \succeq & 0 \end{array}\right\} \tag{10}$$

is "close" to a feasible solution of the DGP, if it exists. This closeness is more precisely defined as follows: given a random $n \times K$ matrix y with each $y_{ij} \sim \mathsf{N}(0, 1/\sqrt{K})$, the $n \times K$ matrix $\bar{x} = \sqrt{\bar{X}}y$ is close to a feasible solution of Eq. (5), in the sense that the Euclidean distance between $\bar{x}$ and each of the submanifolds of $\mathbb{R}^{nK}$ defined by $\|x_i - x_j\|_2 = d_{ij}^2$ is bounded above by $O(\sqrt{\|\bar{X}\|_2\ln(n)})$ (wahp).

Barvinok's naive algorithm also naturally applies to the projected SDP formulation derived from Eq. (6):

$$\left.\begin{array}{rrcl} \forall h \leq k & \sum_{\{i,j\}\in E} T_{h,\{i,j\}}(X_{ii} - 2X_{ij} + X_{jj} - d_{ij}^2) & = & 0 \\ & X & \succeq & 0. \end{array}\right\} \tag{11}$$

Any solution $\bar{X}$ of Eq. (11) should be close (in the sense specified above) to a feasible solution of Eq. (7), i.e. Eq. (8) with zero objective value. We therefore solve Eq. (11), obtain $\bar{X}$, then compute

$\bar{x} = \sqrt{\bar{X}}y$ with $y \sim \mathsf{N}(0, 1/\sqrt{K})^{nK}$, and use it as a starting point for a local NLP solver deployed on Eq. (8): this will yield a solution $\tilde{x}$, which we can then use as a starting point for a local NLP solver deployed on the quartic formulation Eq. (4).

3 Computational assessment

We consider the application of the DGP to the problem of finding the conformation of proteins in space using NMR data, which fixes $K = 3$. Typically, atomic distances up to 5.5Å can be found using NMR experiments and chemical information. We considered three small instances for validation purposes (`lavor30_6-5`, `tiny`, `names`), then derived some realistic instances from the Protein Data Bank (PDB) as follows: we downloaded the protein realization (given with a $O(10^{-3})$ precision), we computed all atomic distances, then ignored all distances larger than 5.5Å. We compare our solution methodology, labelled M_1, with a standard method M_0 consisting in deploying a local NLP solver from a random starting point.

We used Mosek 9.3 [13] in order to solve Eq. (11) (minimizing $\mathsf{trace}(X)$ in an attempt to lead the search towards matrix solutions with low rank), and IPOpt [2] to solve Eq. (3)-(4) locally. The whole testing code, including Barvinok's naive algorithm, was implemented in Python 3.

Instance	n	m	k	mde		lde		rmsd		cpu (sec.)	
				M_0	M_1	M_0	M_1	M_0	M_1	M_0	M_1
$k/m = 0.05$											
`lavor30_6-5`	30	195	10	0.058	**0.028**	**0.560**	0.583	1.857	**1.797**	**0.10**	0.30
`tiny`	37	335	17	0.000	0.000	0.003	**0.000**	2.356	**0.000**	**0.01**	0.04
`names`	86	849	42	0.306	**0.269**	2.889	**2.156**	5.151	**4.082**	**0.23**	1.99
`1guu`	150	955	48	0.118	**0.107**	**1.104**	1.910	**8.569**	9.582	**2.85**	9.06
`1guu-1`	150	959	48	0.092	0.092	**1.096**	1.157	10.217	**7.667**	**0.47**	3.14
$k/m = 0.01$											
`2kxa`	177	2711	27	0.300	**0.281**	3.223	**3.102**	**5.196**	5.770	**1.98**	5.83
`100d`	488	5741	57	0.304	**0.286**	3.689	**3.260**	**9.345**	10.526	**5.50**	25.80
$k/m = 0.005$											
`water`	648	11939	60	0.528	**0.486**	**4.139**	**4.282**	9,423	**8.402**	**11.46**	67.03
`3al1`	678	17417	87	0.320	**0.146**	4.169	**3.312**	**6.712**	7.192	**16.03**	134.01
`1hpv`	1629	18512	93	0.429	**0.425**	3.853	**3.704**	**16.710**	17.178	**48.44**	436.52
$k/m = 0.001$											
`il2`	2084	45251	45	0.163	**0.062**	4.770	**4.220**	12.042	**3.214**	**410.67**	947.92
`1tii`	5684	69800	70	**0.434**	0.453	4.682	**4.327**	26.149	**25.680**	**549.77**	4581.21

Table 1: The test set and results.

The results we obtained are given in Table 1. We indicate the instance name, the number n of vertices, the number m of edges, the number k of constraints in the projected formulation, the mean distance error ($\mathsf{mde}(x) = \mathrm{avg}(|\,\|x_i - x_j\|_2 - d_{ij}\,| \mid \{i,j\} \in E)$), the largest distance error ($\mathsf{lde}(x) = \max(|\,\|x_i - x_j\|_2 - d_{ij}\,| \mid \{i,j\} \in E)$) and the root mean square deviation (rmsd) of the solutions given by methods M_0 (standard) and M_1 (our new method) with respect to the PDB realization, as well as the CPU times (we note, however, that protein conformation is not a time-critical task; and that it is generally preferable to obtain better conformation slowly, than worse conformations fast).

Our results show that the application of RPs to the DGP is successful, and can scale up to large sizes. We are able to retrieve slightly better results than those obtained with the standard method. The fact that it takes longer should not come as a surprise, given that the standard method M_0, aside from the choice of starting point, is essentially the last step of our new algorithm M_1.

References

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

[2] COIN-OR. *Introduction to IPOPT: A tutorial for downloading, installing, and using IPOPT*, 2006.

[3] C. D'Ambrosio, L. Liberti, P.-L. Poirion, and K. Vu. Random projections for quadratic programs. *Mathematical Programming B*, 183:619–647, 2020.

[4] Y. Ding, N. Krislock, J. Qian, and H. Wolkowicz. Sensor network localization, Euclidean distance matrix completions, and graph realization. *Optimization and Engineering*, 11:45–66, 2010.

[5] P. Indyk and J. Matoušek. Low-distortion embeddings of finite metric spaces. In J. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*. Chapman and Hall, Boca Raton, 2004.

[6] W. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In G. Hedlund, editor, *Conference in Modern Analysis and Probability*, volume 26 of *Contemporary Mathematics*, pages 189–206, Providence, RI, 1984. AMS.

[7] L. Liberti. A new distance geometry method for constructing word and sentence vectors. In *Companion Proceedings of the Web Conference (DL4G Workshop)*, volume 20 of *WWW*, New York, 2020. ACM.

[8] L. Liberti. Distance geometry and data science. *TOP*, 28:271–339, 220.

[9] L. Liberti, C. Lavor, N. Maculan, and A. Mucherino. Euclidean distance geometry and applications. *SIAM Review*, 56(1):3–69, 2014.

[10] L. Liberti, P.-L. Poirion, and K. Vu. Random projections for conic programs. *Linear Algebra and its Applications*, 626:204–220, 2021.

[11] L. Liberti and K. Vu. Barvinok's naive algorithm in distance geometry. *Operations Research Letters*, 46:476–481, 2018.

[12] T. Malliavin, A. Mucherino, C. Lavor, and L. Liberti. Systematic exploration of protein conformational space using a distance geometry approach. *Journal of Chemical Information and Modeling*, 59:4486–4503, 2019.

[13] Mosek ApS. *The `mosek` manual, Version 9*, 2019.

[14] M. Pilanci and M. Wainwright. Randomized sketches of convex programs with sharp guarantees. *IEEE Transactions on Information Theory*, 61(9):5096–5115, 2015.

[15] J. Saxe. Embeddability of weighted graphs in k-space is strongly NP-hard. *Proceedings of 17th Allerton Conference in Communications, Control and Computing*, pages 480–489, 1979.

[16] A. Singer. Angular synchronization by eigenvectors and semidefinite programming. *Applied and Computational Harmonic Analysis*, 30:20–36, 2011.

[17] S. Vempala. *The Random Projection Method*. Number 65 in DIMACS Series in Discrete Mathematics and Theoretical Computer Science. AMS, Providence, RI, 2004.

[18] R. Vershynin. *High-dimensional probability*. CUP, Cambridge, 2018.

[19] K. Vu, P.-L. Poirion, and L. Liberti. Random projections for linear programming. *Mathematics of Operations Research*, 43(4):1051–1071, 2018.