Distance geometry: too much is never enough

Gustavo Dias[1] and Leo Liberti[1]

[1]*CNRS LIX, Ecole Polytechnique, 91128 Palaiseau, France.*
{dias,liberti}@lix.polytechnique.fr

Abstract Two years after presenting the distance geometry problem (DGP) as "the most beautiful problem I know" at the last Global Optimization Workshop in Malaga, one of the authors of this abstract (LL) confirms his DGP-mania by proposing a dearth of fun, weird, innovative, elegant and sometimes also practically useful methods for solving this problem, while drawing an unsuspecting Ph.D. student (GD) in the addiction. We present counterintuitive results which only make sense in very high dimensional spaces, adapt the celebrated Isomap heuristic to the DGP setting, and apply some recent techniques for finding feasible solutions of semidefinite programs using a linear programming solver. In short, we do all we can to solve very large DGP instances, albeit approximately.

Keywords: random projections, principal component analysis, diagonally dominant matrix, smoothing

1. Introduction

The Distance Geometry Problem (DGP) consists of "drawing" a weighted graph in a Euclidean space of given dimension, so that a drawn edge is as long as its weight. More precisely, given an integer $K > 0$ and a simple undirected weighted graph $G = (V, E, d)$, where $d : E \to \mathbb{R}_+$, the DGP asks whether there exists a *realization* $x : V \to \mathbb{R}^K$ such that:

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

This problem is **NP**-hard [13] but is not known to be in **NP** [2] for $K > 1$.

A deceptively similar problem called Euclidean Distance Matrix Completion Problem (EDMCP), where K is not given, and the problem asks whether there exists a $K > 0$ such that Eq. (1) holds, is currently not known to be in **P** nor **NP**-hard.

The DGP arises in all applications where one can measure the distances but not the positions of entities: clock synchronization protocols (where $K = 1$ represents the timeline, and one is given time differences but needs to compute absolute clock times), localization of wireless sensors (where $K = 2$ represents e.g. a city block, or an office floor, and pairwise distances are estimated by the amount of battery power consumed in communication), protein conformation (where $K = 3$, and distances are estimated using Nuclear Magnetic Resonance experiments, and the protein binds to a site according to the relative position of its atoms), control of unmanned underwater vehicles (where again $K = 3$, distances are estimated by sonar, and the position cannot be verified directly since GPS signal does not reach underwater). See [8] for more information.

Our favorite method for solving DGPs is Branch-and-Prune (BP) [7]. It scales up to huge sizes [12], is blazingly fast, incredibly accurate [5], polynomial-time "on proteins" [9], and potentially finds all incongruent solutions. But it does not gracefully adapt to distance errors [3] and, most importantly, only works on graphs with a special structure [4]. And so we turn to approximate methods, heuristics, and relaxations.

In this abstract we summarize some of the recent efforts in solving very large DGP instances approximately. We accept approximate solutions because (a) applications usually provide us distances with some errors, and (b) because exact methods do not necessarily scale up to large sizes.

2. Random projections

High dimensional spaces are host to some weird, counterintuitive and somewhat magical-looking phenomena [6]. The one we are specifically interested in is the Johnson-Lindenstrauss Lemma (JLL), which states that if you have a realization x of n points in $\mathbb{R}^K$ and some $\varepsilon \in (0,1)$, then there exists a $k = O((1/\varepsilon^2)\log n)$ and a $k \times K$ matrix T such that:

$$\forall i,j \in V \quad (1-\varepsilon)\|x_i - x_j\|_2^2 \le \|Tx_i - Tx_j\|_2^2 \le (1+\varepsilon)\|x_i - x_j\|_2^2. \tag{2}$$

In fact, if you sample each component of T from $N(0,\sqrt{1/k})$, Eq. (2) holds with probability which approaches 1 exponentially fast as k grows. If you try this out in small dimensional spaces, you will soon see that this is hopeless, which adds a touch of magic to the JLL. We find it even more surprising that the target dimension k is independent of the original dimension K.

Note that the JLL provides a dimensionality reduction mechanism, rather than a solution method for the DGP. Finding a DGP solution in a high dimensional space, however, is easier than finding one with fewer degrees of freedom. So we can project high-dimensional solutions to lower dimensions while keeping the pairwise distances approximately equal. Note that the target dimension k cannot be given: so the JLL applies to the EDMCP rather than the DGP.

Other types of random projections exist, such as Matoušek's, which we also consider.

3. Isomap

The Isomap method [14] is a heuristic method best known for dimensionality reduction, much like the JLL. It works as follows: from a set of n points $X \subseteq \mathbb{R}^K$ we derive a weighted graph $G = (V, E, d)$ from all distances smaller than a given threshold (chosen so as to make the graph connected and reasonably sparse). Note that every edge is weighted with the corresponding Euclidean distance. Next, we complete G to a clique $\bar{G}$ by computing the missing distances using an all (weighted) shortest path algorithm such as Floyd-Warshall. The complete graph $\bar{G}$ is encoded in a symmetric matrix $\bar{D}$ which is an approximation of the (squared) Euclidean Distance Matrix of X. Then we perform classic Multi-Dimensional Scaling (MDS) on $\bar{D}$:

$$\mathcal{G} = -\frac{1}{2} J\bar{D}J, \tag{3}$$

where $\mathcal{G}$ is an approximation of the Gram matrix of X, $J = I - \mathbf{1}/n$, and $\mathbf{1}$ is the all-one $n \times n$ matrix. Since Gram matrices are positive semidefinite (PSD), their eigenvalue matrix Λ has non-negative diagonal, and they can be factored into $YY^\top$ where $Y = P\sqrt{\Lambda}$. $\mathcal{G}$ is not a Gram matrix, however, but only an approximation: so we zero all the negative eigenvalues in Λ (so $\sqrt{\Lambda}$ is real). Finally, we perform a Principal Component Analysis (PCA) step, and discard all but the first K largest eigenvalues of Λ. This yields a set Y of n points in $\mathbb{R}^K$.

Note that Isomap is *almost* a method for solving the DGP. Our "adaptation" consists in a simpe remark: just start Isomap from the weighted graph G.

4. Diagonally dominant programming

In a ground-breaking result, Ahmadi and Hall [1] showed that it is possible to find feasible Semidefinite Programming (SDP) solutions using a Linear Programming (LP) solver. Since

SDP solution technology has a considerable computational bottleneck, this result has the potential for unlocking more SDP power. This result is based on the observation that a diagonally dominant (DD) $n \times n$ matrix $X = (X_{ij})$, namely one such that

$$\forall i \le n \quad X_{ii} \ge \sum_{j \ne i} |X_{ij}|, \tag{4}$$

is also PSD. Note that Eq. (4) can be written linearly by introducing a matrix Y and the constraints:

$$\begin{array}{rrcl} \forall i \le n & \sum\limits_{j \ne i} Y_{ij} & \le & X_{ii} \\ & -Y \le & X & \le Y. \end{array}$$

This means that the PSD constraint $X \succeq 0$ in any SDP can be replaced by the LP constraints above. Programming over those constraints is known as Diagonally Dominant Programming (DDP).

Note that DD implies PSD but not vice-versa. Hence a DDP formulation provides an inner approximation of the SDP feasible region. If the original SDP is used to compute bounds, the guarantee is lost; but since SDP has strong duality, it suffices to apply DDP to the SDP dual. Moreover, the DDP might be infeasible even if the original SDP is feasible. To overcome this issue, Ahmadi and Hall provide an iterative improvement algorithm which enlarges the feasible region of the DDP at each step.

We provide and test DDP formulations for the DGP and EDMCP.

5. The DGSol algorithm

This algorithm was proposed around 20 years ago [10], but it is still very competitive in terms of speed (also thanks to a very good implementation). For smaller scale instances the accuracy of the solutions is not impressive. What is impressive, however, is how well DGSol scales with size in both speed and accuracy. In this sense, DGSol is a truly "big data" kind of method.

The algorithm behind DGSol has an outer and an inner iteration. The outer iteration starts from a smoothed convexified version of the penalty objective function,

$$f(x) = \sum_{\{i,j\} \in E} \left(\|x_i - x_j\|_2^2 - d_{ij}^2 \right)^2$$

obtained via a Gaussian transform

$$\langle f \rangle_\lambda(x) = \frac{1}{\pi^{Kn/2} \lambda^{Kn}} \int\limits_{\mathbb{R}^{Kn}} f(y) \exp(-\|y - x\|_2^2 / \lambda^2) \mathsf{d}y,$$

which tends to $f(x)$ as $\lambda \to 0$.

For each fixed value of λ in the outer iteration, the inner iteration is based on the recursion

$$x^{\ell+1} = x^\ell - \alpha_\ell H_\ell \nabla f(x^\ell),$$

for $\ell \in \mathbb{N}$, where α_ℓ is a step size, and H_ℓ is an approximation of the inverse Hessian matrix of f. In other words, the inner iteration implements a local NLP solution method which uses the optimum at the previous value of λ as a starting point.

Overall, this yields a homotopy method which traces a trajectory depending on $\lambda \to 0$, where a unique (global) optimum of the convex smoothed function $\langle f \rangle_\lambda$ for a high enough value of λ (hopefully) follows the trajectory to the global minimum of the multimodal, nonconvex function $\langle f \rangle_0 = f$.

We use DGSol as a benchmark for comparison. We also borrow its local NLP subsolver for efficiently improving the approximate methods discussed above in a post-processing phase.

6. Conclusion

Our investigations in alternative methods for the DGP are focused towards identifying the best methods for solving very large scale instances of the DGP and EDMCP. Aside from being interesting in their own right, we eventually plan to use them within the BP algorithm in order to provide a better extension for dealing with imprecise data.

Acknowledgments

The first author is financially supported by a CNPq Ph.D. thesis award. The second is partially sponsored by the Bip:Bip ANR project under contract ANR-10-BINF-0003.

References

[1] A. Ahmadi and G. Hall. Sum of squares basis pursuit with linear and second order cone programming. Technical Report 1510.01597v1, arXiv, 2015.

[2] N. Beeker, S. Gaubert, C. Glusa, and L. Liberti. Is the distance geometry problem in **NP**? In Mucherino et al. [11].

[3] A. Cassioli, B. Bordeaux, G. Bouvier, A. Mucherino, R. Alves, L. Liberti, M. Nilges, C. Lavor, and T. Malliavin. An algorithm to enumerate all possible protein conformations verifying a set of distance constraints. *BMC Bioinformatics*, page 16:23, 2015.

[4] C. Lavor, J. Lee, A. Lee-St. John, L. Liberti, A. Mucherino, and M. Sviridenko. Discretization orders for distance geometry problems. *Optimization Letters*, 6:783–796, 2012.

[5] C. Lavor, L. Liberti, N. Maculan, and A. Mucherino. The discretizable molecular distance geometry problem. *Computational Optimization and Applications*, 52:115–146, 2012.

[6] M. Ledoux. *The concentration of measure phenomenon*. Number 89 in Mathematical Surveys and Monographs. American Mathematical Society, Providence, 2005.

[7] L. Liberti, C. Lavor, and N. Maculan. A branch-and-prune algorithm for the molecular distance geometry problem. *International Transactions in Operational Research*, 15:1–17, 2008.

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

[9] L. Liberti, C. Lavor, and A. Mucherino. The discretizable molecular distance geometry problem seems easier on proteins. In Mucherino et al. [11].

[10] J. Moré and Z. Wu. Global continuation for distance geometry problems. *SIAM Journal of Optimization*, 7(3):814–846, 1997.

[11] A. Mucherino, C. Lavor, L. Liberti, and N. Maculan, editors. *Distance Geometry: Theory, Methods, and Applications*. Springer, New York, 2013.

[12] A. Mucherino, C. Lavor, L. Liberti, and E-G. Talbi. A parallel version of the branch & prune algorithm for the molecular distance geometry problem. In *ACS/IEEE International Conference on Computer Systems and Applications (AICCSA10)*, pages 1–6, Hammamet, Tunisia, 2010. IEEE.

[13] 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.

[14] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2322, 2000.