

RESEARCH

Open Access

Structure method for solving the nearest Euclidean distance matrix problem

Suliman Al-Homidan*

*Correspondence:
homidan@kfupm.edu.sa
Department of Mathematics and
Statistics, King Fahd University of
Petroleum and Minerals, Dhahran,
Saudi Arabia

Abstract

A matrix with zero diagonal is called a Euclidean distance matrix when the matrix values are measurements of distances between points in a Euclidean space. Because of data errors such a matrix may not be exactly Euclidean and it is desirable in many applications to find the best Euclidean matrix which approximates the non-Euclidean matrix. In this paper the problem is formulated as a smooth unconstrained minimization problem, for which rapid convergence can be obtained. Comparative numerical results are reported.

Keywords: Euclidean distance matrix; positive semidefinite matrix; Newton method; BFGS method

1 Introduction

Symmetric matrices with non-negative off-diagonal elements and zero diagonal elements arise as data in many experimental sciences. This occurs when the values are measurements of squared distances between points in a Euclidean space (e.g. atoms, stars, cities). Such a matrix is referred to as a *Euclidean distance matrix*. Because of data errors such a matrix may not be exactly Euclidean and it is desirable to find the best Euclidean matrix which approximates the non-Euclidean matrix. The aim of this paper is to study a new method for solving the Euclidean distance matrix problem and compare it with other older methods [1].

An important application arises in the conformation of molecular structures from nuclear magnetic resonance data (see [2] and [3]). Here a Euclidean distance matrix is used to represent the squares of distances between the atoms of a molecular structure. An attempt to determine such a structure by nuclear magnetic resonance experiments gives rise to a distance matrix F which, because of data errors, may not be Euclidean. There are many other applications in subjects as diverse as archeology, cartography, genetics, geography, and multivariate analysis. Pertinent references are given by Al-Homidan [4, 5].

Characterization theorems for the Euclidean distance matrix have been given in many forms. In Section 2 we show a very important characterization which brings out the underlying structure and is readily applicable to the algorithms that follow.

This paper addresses a non-smooth optimization problem in which some matrix, defined in terms of the problem variables, has to be positive semidefinite. One way to handle this problem is to impose a functional constraint in which the least eigenvalue of the matrix is non-negative. However, if there are multiple eigenvalues at the solution, which is

usually the case, such a constraint is non-smooth, and this non-smoothness cannot be modeled by a convex polyhedral composite function. An important factor is the determination of the multiplicity of the zero eigenvalues, or alternatively the *rank* of the matrix at the solution. If this rank is known it is usually possible to solve the problem by conventional techniques.

Glunt *et al.* [6] formulate the Euclidean distance matrix problem as a constrained least distance problem in which the constraint is the intersection of two convex sets. The Dykstra-Han alternating projection algorithm can then be used to solve the problem. This method is globally convergent but the rate of convergence is very slow. However, the method does have the capability to determine the correct rank of the solution matrix.

Recently, there has been much interest in the interior point methods applied to problems with semidefinite matrix constraints (*e.g.* the survey papers [7] and [8] and the references therein). Semidefinite programming optimizes a linear function subject to a positive semidefinite matrix. It is a convex programming problem since the objective and constraints are convex. In this paper, we deal with a problem that is a little different since the objective is quadratic; also an additional rank constraint is added which makes the problem non-convex and harder to solve. Here, we use a different approach from the interior point methods. If the correct rank of the solution matrix is known, it is shown in Section 3 how to formulate the problem as a smooth unconstrained minimization problem, for which rapid convergence can be obtained by for example the BFGS method. We give expressions for the objective function and its first derivatives.

In [1] a hybrid method is studied between a projection method and a quasi-Newton method; a similar study can be performed as regards all its features. Finally, in Section 4, numerical comparisons are carried out.

2 The Euclidean distance matrix problem

In this section the definition of the Euclidean distance matrix is given, and the relationship between points and distances is summarized. A characterization theorem for the Euclidean distance matrix is proved in a concise way that brings out the underlying structure and is readily applicable to the algorithms that follow.

It is necessary to distinguish between distance matrices that are obtained in practice and those that can be derived exactly from n vectors in an affine subspace.

Definition 2.1 A matrix $F \in \mathbb{R}^{n \times n}$ is called a distance matrix iff it is symmetric, the diagonal elements are zero

$$f_{ii} = 0, \quad i = 1, \dots, n,$$

and the off-diagonal entries are non-negative

$$f_{ij} \geq 0, \quad \forall i \neq j.$$

Definition 2.2 A matrix $D \in \mathbb{R}^{n \times n}$ is called a Euclidean distance matrix iff there exist n vectors $\mathbf{x}_1, \dots, \mathbf{x}_n$ in an affine subspace of dimension \mathbb{R}^r ($r \leq n - 1$) such that

$$d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2^2, \quad \forall i, j. \tag{2.1}$$

The Euclidean distance problem can now be stated as follows: Given a distance matrix $F \in \mathbb{R}^{n \times n}$, find the Euclidean distance matrix $D \in \mathbb{R}^{n \times n}$ that minimizes

$$\|F - D\|_F, \tag{2.2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

The theorem is essentially due to Schoenberg [9].

Theorem 2.3 *The distance matrix $D \in \mathbb{R}^{n \times n}$ is a Euclidean distance matrix if and only if the $(n - 1) \times (n - 1)$ symmetric matrix A defined by*

$$a_{ij} = \frac{1}{2}[d_{s1} + d_{t1} - d_{st}] \quad (1 \leq i, j \leq n - 1) \tag{2.3}$$

is positive semidefnite, where $s = i + 1$, $t = j + 1$, and D is irreducibly embeddable in \mathbb{R}^r ($r < n$) where $r = \text{rank}(A)$. Moreover, consider the spectral decomposition

$$A = U \Lambda U^T. \tag{2.4}$$

Let Λ_r be the matrix of non-zero eigenvalues in Λ and define X by

$$X = U_r, \quad \text{then } A = X \Lambda_r X^T, \tag{2.5}$$

where $\Lambda_r \in \mathbb{R}^{r \times r}$ is a diagonal matrix and $U_r \in \mathbb{R}^{(n-1) \times r}$.

3 The method

In this section we consider a different approach to the Euclidean distance matrix problem (2.2). The main idea is to replace (2.2) by a smooth unconstrained optimization problem in order to use superlinearly convergent quasi-Newton methods. To do this it is necessary to estimate the rank r as this piece of information is not generally known. Once a value of r is chosen, the problem (2.2) is solved by the BFGS method. We give the relevant formulas for the derivatives. At the end of the section we discuss details of the initialization and implementation.

If the rank r is known, it is possible to express (2.2) as a smooth unconstrained optimization problem in the following way. The unknowns in the problem are chosen to be the elements of the matrix X and Λ_r introduced in (2.5). We take X to have r columns and Λ_r a diagonal matrix as shown below. This gives us an unconstrained optimization problem in $r(n - 1) - \frac{r(r+1)}{2}$ unknowns. We therefore parametrize X and Λ_r in the following way:

$$A = X \Lambda_r X^T, \quad \text{where } X = \begin{bmatrix} 1 & 0 & 0 \\ x_{21} & 1 & 0 \\ \vdots & x_{32} & 1 \\ \vdots & \vdots & x_{r+1,r} \\ \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & x_{mr} \end{bmatrix}, \quad \Lambda_r = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_r \end{bmatrix}. \tag{3.1}$$

The objective function $\phi(X)$ is readily calculated by first forming D from X and Λ_r as indicated by (2.1), after which ϕ is given by $\phi(X, \Lambda_r) = \|D - F\|_F^2$. When $s = t$, then $d_{st} = 0$, using (2.3) we get $a_{ii} = \frac{1}{2}[d_{s1} + d_{s1} - 0] = d_{s1}$, then the elements of the matrix D take the form

$$d_{s1} = a_{ii} = \sum_{k=1}^{i-1} x_{ik}^2 \lambda_k + \lambda_i \quad \text{if } i = 1, \dots, r, s = i + 1,$$

$$d_{s1} = a_{ii} = \sum_{k=1}^r x_{ik}^2 \lambda_k \quad \text{if } i = r + 1, \dots, n - 1,$$

$$d_{st} = a_{ij} = \sum_{k=1}^{i-1} x_{ik} x_{jk} \lambda_k + x_{ij} + \lambda_i \quad \text{if } i \leq r \text{ or } j \leq r,$$

$$d_{st} = a_{ij} = \sum_{k=1}^r x_{ik} x_{jk} \lambda_k \quad \text{if } i > r \text{ and } j > r,$$

where $t = j + 1$. Hence

$$\begin{aligned} \phi &= \sum_{s,t=1}^n (d_{ij} - f_{ij})^2 = 2 \sum_{s=2}^n (d_{s1} - f_{s1})^2 + 2 \sum_{\substack{s,t=2 \\ s < t}}^n (d_{st} - f_{st})^2 \\ &= 2 \sum_{i=1}^r \left[\sum_{k=1}^{i-1} x_{ik}^2 \lambda_k + \lambda_i - f_{s1} \right]^2 + 2 \sum_{i=r+1}^{n-1} \left[\sum_{k=1}^r x_{ik}^2 \lambda_k - f_{s1} \right]^2 \\ &\quad + 2 \sum_{\substack{i,j=1 \\ i < j}}^r \left[\sum_{k=1}^{i-1} x_{ik}^2 \lambda_k + \lambda_i + \sum_{k=1}^{j-1} x_{kj}^2 \lambda_k + \lambda_j - 2 \sum_{k=1}^{i-1} x_{ik} x_{kj} \lambda_k + x_{ij} \lambda_i - f_{st} \right]^2 \\ &\quad + 2 \sum_{i=1}^r \sum_{j=r+1}^{n-1} \left[\sum_{k=1}^{i-1} x_{ik}^2 \lambda_k + \lambda_i + \sum_{k=1}^r x_{kj}^2 \lambda_k - 2 \sum_{k=1}^{i-1} x_{ik} x_{kj} \lambda_k + x_{ij} \lambda_i - f_{st} \right]^2 \\ &\quad + 2 \sum_{\substack{i,j=r+1 \\ i < j}}^{n-1} \left[\sum_{k=1}^r x_{ik}^2 \lambda_k + \sum_{k=1}^r x_{kj}^2 \lambda_k - 2 \sum_{k=1}^r x_{ik} x_{kj} \lambda_k - f_{st} \right]^2. \end{aligned} \tag{3.2}$$

Our chosen method to minimize $\phi(X)$ is the BFGS quasi-Newton method (see for example [10]). This requires expressions for the first partial derivatives of ϕ , which are given from (3.2) by

$$\begin{aligned} \frac{\partial \phi}{\partial \lambda_i} &= 2 \left\{ 2 \sum_{l=1}^i \sum_{k=l+i}^{n-1} (d_{k+l} - f_{k+l}) x_{ik}^2 + 2 \sum_{k=i+1}^{n-1} (d_{k+1+i} - f_{k+1+i}) (1 + x_{ik}^2 - 2x_{ik}) \right. \\ &\quad \left. + 2 \sum_{l=i+2}^r \sum_{k=l}^{n-1} (d_{k+l} - f_{k+l}) (x_{il-1}^2 + x_{ik}^2 - 2) x_{i,l-1} x_{ik} \right\}, \end{aligned} \tag{3.3}$$

for all $i = 1, \dots, r$. For $j = 1, \dots, r$, and $i = j + 1, \dots, n - 1$:

$$\frac{\partial \phi}{\partial x_{ij}} = 4 \sum_{k=0}^{i-1} (d_{j+1k+1} - f_{j+1k+1})(2x_{ij}\lambda_j) + 4 \sum_{k=i}^{n-1} (d_{j+1k+1} - f_{j+1k+1})(2x_{ij}\lambda_j - 2x_{kj}\lambda_j). \quad (3.4)$$

The BFGS method also requires the Hessian approximation to be initialized. Where necessary, we do this using a unit matrix.

Some care has to be taken when choosing the initial value of the matrix X and Λ_r , in particular the rank must be r . If not, the minimization method may not be able to increase the rank of X . An extreme case occurs when the initial matrix $X = 0$ and $\Lambda_r = 0$ are chosen, and $F \neq 0$. It can be seen from (3.3) and (3.4) that the components of the gradient vector are all zero, so that $X = 0$ and $\Lambda_r = 0$ are stationary points, but not minimizers. A gradient method will usually terminate in this situation and so fail to find the solution.

A reliable method for initializing X and $\Lambda_r = 0$ is to use the construction suggested by (3.1) and (2.3). Thus we define the elements of A by those of F by

$$a_{ij} = \frac{1}{2}(f_{ij} - f_{i1} - f_{1j}), \quad i \geq 2, j \geq 2. \quad (3.5)$$

The first row and column of A are zero and are ignored. We then find the spectral decomposition $U\Sigma U^T$ of the nontrivial part of A . Finally the nontrivial part of X and Λ_r in (3.1) is initialized to the matrix $\Sigma_r^{1/2}U_r^T$ where $\Sigma_r = \text{diag}(\sigma_i)$, $i = 1, \dots, r$ is composed of the r largest eigenvalues in Σ , and the columns of U_r are the corresponding eigenvectors. When Σ_r is positive definite, this procedure ensures that A has the correct rank r . Otherwise the process must be modified in some way, for example by ensuring that the diagonal elements in Σ_r lie above a positive threshold.

An advantage of this method is that it allows the spatial dimensions to be chosen by the user. This is useful when the rank is already known. For example if the entries in F are derived from distances between cities then the dimension will be no higher than $r = 2$. Likewise, if the entries are derived from distances between atoms in a molecule or stars in space, then the maximum dimension is $r = 3$.

In general, however, the rank is not known, for example the atoms in a molecule may turn out to be collinear or coplanar. We therefore must consider an algorithm in which we are prepared to revise our estimate of r . A simple strategy is to repeat the entire method for different values of r . If r^* denotes the correct value of r which solves (2.2), then it is observed that the BFGS method converges rapidly if $r \leq r^*$, and that it exhibits superlinear convergence. On the other hand if $r > r^*$ then slow convergence is observed. One reason is that there are more variables in the problem. Also redundancy in the parameter space may have an effect. Thus it makes sense to start with a small value of r , and increase it by one until the solution is recognized. One way to recognize termination is when $D^{(r)}$ agrees sufficiently well with $D^{(r+1)}$, where $D^{(r)}$ denotes the Euclidean distance matrix obtained by minimizing ϕ when Λ_r in (3.1) has r diagonal elements. Numerical experience is reported in [4] for solving various test problems by other methods which will be compared with this method.

An obvious alternative to using the BFGS method is to evaluate the Hessian matrix of second derivatives of $\phi(X)$ and use Newton's method. This would likely reduce the number of iterations required. However, there is also the disadvantage of increased complexity, and increased housekeeping at each iteration. Moreover, it is possible that the Hessian

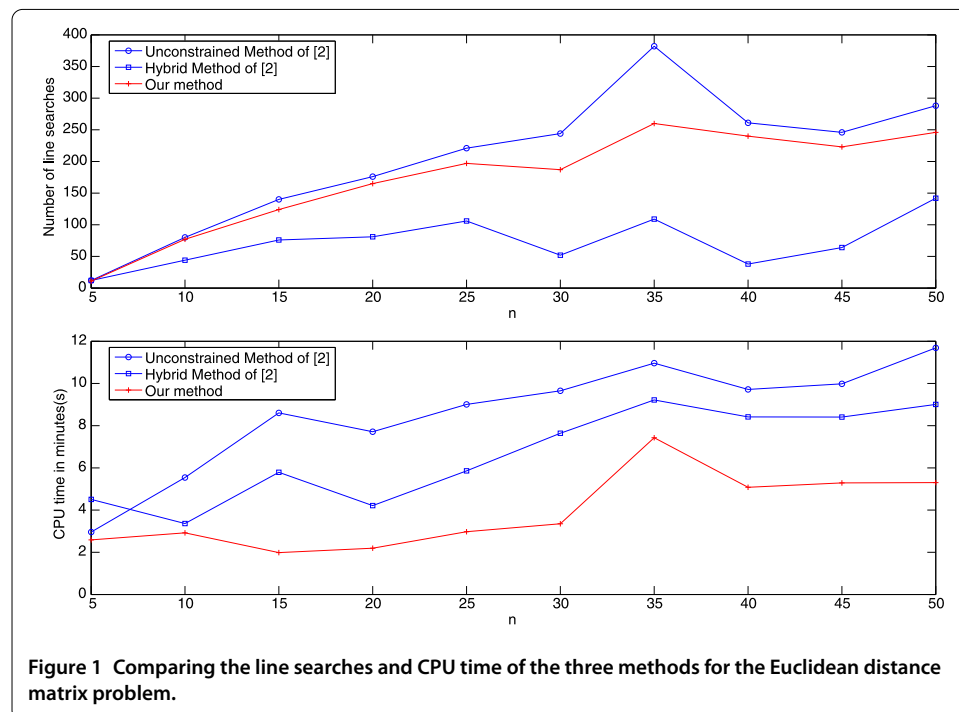
has some negative eigenvalues so a modified form of Newton’s method would be required. A simple example serves to illustrate the possibility of a negative eigenvalue. Take $n = 2$, $r = 1$, and let $F = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$, $X = [1]$, and $\Lambda_r = [\lambda_1]$. Then $\phi = 2(1 - \lambda_1^2)^2$. This has global minimizers at $\lambda_1 = \pm 1$, a local maximizer at $\lambda_1 = 0$, and the Hessian is negative for all λ_1 such that $3\lambda_1^2 < 1$.

This method has entirely different features, some good, some bad, which suggests that a combination of both this method and a projection method [6] might be successful. Projection methods are globally convergent and hence potentially reliable, but the rate of convergence is first order or slower, which can be very inefficient. Quasi-Newton methods are reliable and locally superlinearly convergent, but they require that the correct rank r^* is known. Therefore hybrid methods should be established along the lines of [1], in which the projection algorithm is used sparingly as a way of establishing the correct rank, while the BFGS method is used to provide rapid convergence.

4 Numerical results

In this section, we compare three methods, our method, the hybrid method in [1] and the unconstrained method of the same reference. The algorithms have been tested on randomly generated distance matrices F with values distributed between 10^{-3} and 10^3 . All calculations were performed with Matlab 8. Figure 1 compares the line searches and CPU time of the three methods. The termination criterion for both methods is $\|D^{(k)} - D^{(k-1)}\| < 10^{-5}$. All methods converge to essentially the same values.

In Figure 1, the upper figure shows that the number of line searches for our method is slightly lower than the unconstrained method and higher than the hybrid method. However, in the lower figure it is clear that our method is much faster and this because our method has $\frac{r(r+1)}{2}$ less CPU time. A hybrid method uses much less line searches from both



methods, however, it consumes much more time than our method because it uses a projection method as a start. This makes our method more efficient and faster.

The housekeeping associated with each line search is $O(n^2)$. Also, if care is taken, it is possible to calculate $\phi(X)$ and $\nabla\phi(X)$ in $O(n^2)$ operations. The initial value $r^{(0)}$ is tabulated, and r is increased by one until the solution is found. The total number of line searches is tabulated, and in this figure, it is found that fewer line searches are required as r increases. Also the initial value $r^{(0)} = 6$ is rather arbitrary: a smaller value of $r^{(0)}$ would have given an even larger number of line searches.

Competing interests

The author declares that they have no competing interests.

Acknowledgements

The author is grateful to King Fahd University of Petroleum & Minerals for providing excellent research facilities.

Received: 2 September 2014 Accepted: 24 November 2014 Published: 12 Dec 2014

References

1. Al-Homidan, S, Fletcher, R: Hybrid methods for finding the nearest Euclidean distance matrix. In: Du, D, Qi, L, Womersley, R (eds.) *Recent Advances in Nonsmooth Optimization*, pp. 1-17. World Scientific, Singapore (1995)
2. Havel, T, Kuntz, I, Crippen, GM: The theory and practice of distance geometry. *Bull. Math. Biol.* **45**, 665-720 (1983)
3. Crippen, GM: Rapid calculation of coordinates from distance measures. *J. Comput. Phys.* **26**, 449-452 (1978)
4. Al-Homidan, S: Hybrid methods for optimization problems with positive semi-definite matrix constraints. Ph.D. thesis, Dept. of Mathematics and Computer Science, University of Dundee, Dundee, Scotland (1993)
5. Al-Homidan, S, Fletcher, R: Rationalizing foot and ankle measurements to conform to a rigid body model. *Comput. Methods Biomech. Biomed. Eng.* **9**(2), 103-111 (2006)
6. Glunt, W, Hayden, TL, Hong, S, Wells, J: An alternating projections method for computing the nearest Euclidean distance matrix. *SIAM J. Matrix Anal. Appl.* **4**, 589-600 (1990)
7. Al-Homidan, S, Wolkowicz, H: Approximate and exact completion problems for Euclidean distance matrices using semidefinite programming. *Linear Algebra Appl.* **406**, 109-141 (2005)
8. Wolkowicz, H, Saigal, R, Vandenberghe, L: *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*. Kluwer Academic, Dordrecht (2000)
9. Schoenberg, IJ: Remarks to M. Frechet's article 'Sur la definition axiomatique d'une classe d'espace distances vectoriellement applicable sur l'espace de Hilbert'. *Ann. Math.* **36**, 724-732 (1935)
10. Fletcher, R: *Practical Methods of Optimization*, 2nd. edn. Wiley, Chichester (1987)

10.1186/1029-242X-2014-491

Cite this article as: Al-Homidan: Structure method for solving the nearest Euclidean distance matrix problem. *Journal of Inequalities and Applications* 2014, **2014**:491

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com