Lecture 3.5: Geometric Graph Embeddings with Partial Data

Radu Balan

Department of Mathematics, AMSC, CSCAMM and NWC
University of Maryland, College Park, MD

February 13, 2025

Embedding Problems

Problem Statement and Ambiguities

Main Problem

Isometric Embedding: Given the set of all squared-distances $\{d_{i,j}^2;\ 1 \leq i,j \leq n\}$ *find a dimension* d *and a set of* n *points* $\{y_1, \cdots, y_n\} \subset \mathbb{R}^d$ *so that* $\|y_i - y_j\|^2 = d_{i,j}^2$, $1 \leq i,j \leq n$.

Main Problem

Nearly Isometric Embedding: Given the set of all squared-distances $\{d_{i,j}^2;\ 1 \leq i,j \leq n\}$ *find a dimension* d *and a set of* n *points* $\{y_1, \cdots, y_n\} \subset \mathbb{R}^d$ *so that* $\|y_i - y_j\|^2 \approx d_{i,j}^2$, $1 \leq i,j \leq n$.

Note the set of points is unique up to rigid transformations: translations, rotations and reflections: $\mathbb{R}^d \times O(d)$. This means two sets of n points in $\mathbb{R}^d$ have the same pairwise distances if and only if one set is obtained from the other set by a combination of rigid transformations.

Isometric Embeddings with Partial Data

Dimension estimation

Consider now the case that only a subset of the pairwise squared-distances are known, indexed by Θ. Assume that only m distances (out of $n(n-1)/2$ possible values) are known – this means the cardinal of Θ is m.

Remark

Minimum number of measurements: $m \geq nd - \frac{d(d+1)}{2}$, *because:* nd *is the number of degrees of freedom (coordinates) needed to describe n points in* $\mathbb{R}^d$; $d(d+1)/2$ *is the the dimension of the Lie group of Euclidean transformations: translations in* $\mathbb{R}^d$ *of dimension* d *and orthogonal transformations* $O(d)$ *of dimension* $d(d-1)/2$ *(the dimension of the Lie algebra of anti-symmetric matrices).*

In the absence of noise, for sufficiently large m but less than $n(n-1)/2$, exact (i.e. isometric) embedding is possible.

Geometry of the (Lie) Group $O(d)$

Recall the definition of orthogonal matrices: A matrix $U \in \mathbb{R}^{d\times d}$ is called *orthogonal* if $UU^T = I_d$. Note this means the matrix U is invertible, $U^{-1} = U^T$ and therefore $U^T U = I_d$. Hence if U is an orthogonal matrix so is U^T.

Let $O(n)$ denote the set of all $d \times d$ orthogonal matrices. Notice that it satisfies the following properties:

1. $I_d := eye(d)$ is an orthogonal matrix, $I_d \in O(d)$;
2. If $U \in O(d)$ then $U^T \in O(d)$ and $UU^T = U^T U = I_d$;
3. If $U, V, W \in O(d)$ then:
$$(UV)W = U(VW)$$
4. If $U, V \in O(d)$ then $UV \in O(d)$ because:
$$(UV)(UV)^T = UVV^T U^T = UU^T = I_d$$

All these properties combined say that $(O(d), \cdot)$ forms a *group*. Here $\cdot$ denotes the matrix multiplication.

In addition to abstract algebraic properties, the $O(d)$ group admits more analytical and geometric properties. All these make $O(d)$ a prime example of a *Lie group*. Specifically:

1. the set $O(d)$ has the structure of a *manifold* (generalization of the concepts of "curve" and "surface" from $\mathbb{R}^3$);
2. the matrix multiplication and inversion are differentiable maps.

In addition to abstract algebraic properties, the $O(d)$ group admits more analytical and geometric properties. All these make $O(d)$ a prime example of a *Lie group*. Specifically:

1. the set $O(d)$ has the structure of a *manifold* (generalization of the concepts of "curve" and "surface" from $\mathbb{R}^3$);
2. the matrix multiplication and inversion are differentiable maps.

Two properties of matrix determinant:
i) For any $A, B \in \mathbb{R}^{d\times d}$, $det(AB) = det(A)det(B)$.
ii) For any $A \in \mathbb{R}^{d\times d}$, $det(A^T) = det(A)$.
This implies: for any $U \in O(d)$,

$$1 = det(I) = det(UU^T) = det(U)det(U^T) = (det(U))^2$$

Thus $det(U) = \pm 1$. We define:

$$SO(d) = \{U \in O(d)\,;\ det(U) = 1\} = \{U \in \mathbb{R}^{d\times d}\ ,\ \ UU^T = I\,,\ det(U) = 1\}$$

called the *special orthogonal group of order d*.

$SO(d)$ represents the connected component of $O(d)$, that is, the set of orthogonal matrices that can be connected by a continuous path to the identity. As we shall see later, the continuous path can be constructed using the matrix exponential map. The complement set $O(d) \setminus SO(d)$ is also a connected component (but not a subgroup of $O(d)$).
Consider a differentiable path $\gamma : (-1, 1) \to SO(d)$, $\gamma(0) = I$. We want to find the tangent vector of this curve at $t = 0$. The set of such vectors is called the *tangent space* to manifold $SO(d)$ (and implicitly to manifold $O(d)$). We denote this tangent space by $so(d)$.
Let's compute them:

$$\gamma(t)\gamma(t)^T = I \to \frac{d}{dt}\left(\gamma(t)\gamma(t)^T\right)|_{t=0} = 0$$

Using the product rule and the fact that $\gamma(0) = I$, the above identity reduces to:

$$\frac{d\gamma(t)}{dt}(0) + \frac{d\gamma(t)}{dt}(0)^T = 0.$$

Hence:

$$so(d) = \{A \in \mathbb{R}^{d\times d} \quad , \quad A + A^T = 0\}$$

is the set of anti-symmetric matrices. We are going to use this information (the tangent space) to determine the *dimension* of the group $O(d)$, or $SO(d)$.

First, notice the following properties:

1. $so(d)$ is a vector space: if A, B are anti-symmetric matrices so is $A + B$ as well as cA, for anay $c \in \mathbb{R}$.
2. Since $so(d)$ is a vector space, subspace of $\mathbb{R}^{d\times d}$, it has a finite dimension. Let $p = dim(so(d))$. Since all anti-symmetric matrices have 0 on the main diagonal, and the upper elements are repeated on the lower half of the matrix, with sign changed, the dimension of $so(d)$ must be

$$p = dim(so(d)) = \frac{d(d-1)}{2}$$

In addition to the vector space structure, $so(d)$ has an additional internal operation, the *Lie bracket* (or the *commutator*):

$$A, B \in so(d) \to [A, B] = AB - BA \in so(d)$$

It is bilinear, anti-symmetric and satisfies a 3-term identity (called the Jacobi identity): for every $A, B, C \in so(d)$, $\alpha, \beta, \gamma \in \mathbb{R}$,

1. $[\alpha A + \beta B, C] = \alpha[A, C] + \beta[B, C]$, $[A, \beta B + \gamma C] = \beta[A, B] + \gamma[A, C]$;
2. $[A, A] = 0$
3. $[A, [B, C]] + [B, [C, A]] + [C, [A, B]] = 0$

These tree properties define a *Lie algebra*. Thus $so(d)$ is a Lie algebra of dimension $\frac{d(d-1)}{2}$.
In general any Lie group $(G, \cdot)$ admits a Lie algebra $(g, +, [,])$ of some dimension p. The converse is also true (one of Lie theorems).

Isometric Embeddings with Partial Data

Linear constraints

Given any set of vectors $\{y_1, \cdots, y_n\}$ and their associated matrix $Y = [y_1|\cdots|y_n]$ their invariant to the action of the rigid transformations (translations, rotations, and reflections) is the Gram matrix of the centered system (L is an orthogonal projection):

$$G = (I - \frac{1}{n}1 \cdot 1^T)Y^T Y(I - \frac{1}{n}1 \cdot 1^T) =: LY^T YL \quad , \quad L = I - \frac{1}{n}1 \cdot 1^T.$$

On the other hand, the distance between points i and j can be computed by:

$$d_{i,j}^2 = \|y_i - y_j\|^2 = G_{i,i} - G_{i,j} + G_{j,j} - G_{j,i} = e_{ij}^T G e_{ij}$$

where

$$e_{ij} = \delta_i - \delta_j = [0 \cdots 0\, 1 \cdots - 1\, 0 \cdots 0]^T$$

where 1 is on position i, -1 is on position j, and 0 everywhere else.

Almost Isometric Embeddings with Partial Data

The SDP Problem

Reference [3] proposes to find the matrix G by solving the following Semi-Definite Program:

$$\min_{\substack{G=G^T\geq 0 \\ G1=0 \\ |\langle Ge_{ij},e_{ij}\rangle - \tilde{d}^2_{i,j}|\leq\varepsilon\ ,\ (i,j)\in\Theta}} \quad trace(G)$$

where $\tilde{d}^2_{i,j}$ are noisy estimates $d_{i,j}$ and ε is the maximum noise level. The trace promotes low rank in this optimization. Overall this is a feasibility problem: Decrease ε to the minimum value where a feasible solution exists. With probability 1 that is unique.
How to do this: Use CVX for Matlab. Procedure summarized in Alg 3 and Alg 4 next.

Nearly Isometric Embeddings with Partial Data

Stability to Noise

Let $\Theta_r = \{(i,j)\ ,\ \|y_i - y_j\| \leq r\}$ be the set of all pairs of points at distance at most r.

Theorem (Javanmard, Montanari[3])

Let $\{y_1, \cdots, y_n\}$ be n nodes distributed uniformly at random in the hypercube $[-0.5, 0.5]^d$. Further, assume that we are given noisy measurement of all distances in Θ_r for some $r \geq 10\sqrt{d}(\log(n)/n)^{1/d}$ and the induced geometric graph of edges is connected. Let $\tilde{d}^2_{i,j} = d^2_{i,j} + \nu_{i,j}$ with $|\nu_{i,j}| \leq \varepsilon$. Then with high probability, the error distance between the estimated $\hat{Y} = [\hat{y}_1, |\cdots|\hat{y}_n]$ returned by the SDP algorithm and the true coordinate matrix $Y = [y_1|\cdots|y_n]$ is upper bounded as

$$\|L\hat{Y}^T\hat{Y}L - LY^TYL\|_1 \leq C_1(nr^d)^5\frac{\varepsilon}{r^4}.$$

Conversely, w.h.p., there exist adversarial measurement errors $\{z_{i,j}\}_{(i,j)\in\Theta_r}$ such that

$$\|L\hat{Y}^T\hat{Y}L - LY^TYL\|_1 \geq C_2 min(1, \frac{\varepsilon}{r^4}).$$

Here, C_1 and C_2 denote universal constants that depend only on d, and $L = I - \frac{1}{n}1\cdot 1^T$.

Computation of the Gram matrix

Algorithm (Alg 3 - The SDP Problem)

Inputs:Collection of squared pairwise distances $\mathbb{S} = \{d_{i,j}^2\ ,\ (i,j) \in \Theta\}$*; noise level parameters:*$\varepsilon_0, \varepsilon_{min} > 0$*; (optional) maximum number of iterations:*N_{max}*.*

1. *Initialize* $\varepsilon = \varepsilon_0$*. If a maximum number of iterations is used, initialize* $k = 1$*.*
2. *Solve:*

$$\min_{\substack{G = G^T \geq 0 \\ G1 = 0 \\ |\langle Ge_{ij}, e_{ij}\rangle - \tilde{d}_{i,j}^2| \leq \varepsilon\ ,\ (i,j) \in \Theta}} trace(G) \quad (SDP)$$

Computation of the Gram matrix -cont'd

Algorithm (Alg 3 - continued)

3. *If no solution of the SDP is found:*
   a. *if a solution was found at previous iteration, then report that solution G, parameter ε and the iteration index k for that solution;*
   b. *if no solution found so far, then increase ε_0, for instance double $\varepsilon_0 = 2\varepsilon_0$, and go back to step 1.*

   Else, if a solution of the SDP is found:
   a. *If $\varepsilon > \varepsilon_{min}$ and (optional) $k < N_{max}$ then decrease ε, e.g., $\varepsilon = \varepsilon/2$, increment $k = k + 1$, and then go back to step 2.*
   b. *Else, report the last solution found G and parameters ε, and number of iterations k.*

Output: Symmetric Gram matrix G, parameter ε, number of iterations k.

Gram matrix factorization

Algorithm (Alg 4)

Input: Symmetric $n \times n$ Gram matrix G.

1. *Compute the eigendecomposition of G, $G = Q\Lambda Q^T$ with diagonal of Λ sorted in a descending order;*
2. *Determine the number d of significant positive eigevalues;*
3. *Partition*
$$Q = [Q_1 \quad Q_2] \quad \text{, and} \quad \Lambda = \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix}$$
where Q_1 contains the first d columns of Q, and Λ_1 is the $d \times d$ diagonal matrix of significant positive eigenvalues of G.
4. *Compute:*
$$Y = \Lambda_1^{1/2} Q_1^T$$

Output: Dimension d and $d \times n$ matrix Y of vectors $Y = [y_1 | \cdots | y_n]$

Convex Sets. Convex Functions

A set $S \subset \mathbb{R}^n$ is called a *convex set* if for any points $x, y \in S$ the line segment $[x, y] := \{tx + (1-t)y \ , \ 0 \le t \le 1\}$ is included in S, $[x, y] \subset S$.

A function $f : S \to \mathbb{R}$ is called *convex* if for any $x, y \in S$ and $0 \le t \le 1$, $f(t\,x + (1-t)y) \le t\,f(x) + (1-t)f(y)$.
Here S is supposed to be a convex set in $\mathbb{R}^n$.
Equivalently, f is convex if its epigraph is a convex set in $\mathbb{R}^{n+1}$. Epigraph: $\{(x, u) \ ; \ x \in S, u \ge f(x)\}$.

A function $f : S \to \mathbb{R}$ is called *strictly convex* if for any $x \neq y \in S$ and $0 < t < 1$, $f(t\,x + (1-t)y) < t\,f(x) + (1-t)f(y)$.

Convex Optimization Problems

The general form of a convex optimization problem:

$$\min_{x \in S} f(x)$$

where S is a closed convex set, and f is a convex function on S.

Properties:

1. Any local minimum is a global minimum. The set of minimizers is a convex subset of S.
2. If f is strictly convex, then the minimizer is unique: there is only one local minimizer.

In general S is defined by equality and inequality constraints:
$S = \{g_i(x) \leq 0 \ , \ 1 \leq i \leq p\} \cap \{h_j(x) = 0 \ , \ 1 \leq j \leq m\}$. Typically h_j are required to be affine: $h_j(x) = a^T x + b$.

Convex Programs

The hiarchy of convex optimization problems:

1. Linear Programs: Linear criterion with linear constraints
2. Quadratic Programs: Quadratic Criterion with Linear Constraints; Quadratically Constrained Quadratic Problems (QCQP); Second-Order Cone Program (SOCP)
3. Semi-Definite Programs(SDP)

Typical SDP:

$$\min_{\substack{X = X^T \geq 0 \\ trace(XB_k) = y_k \ , \ 1 \leq k \leq p \\ trace(XC_j) \leq z_j \ , \ 1 \leq j \leq m}} trace(XA)$$

CVX

Matlab package

Downloadable from: http://cvxr.com/cvx/ . Follows "Disciplined" Convex Programming – à la Boyd [1].

```
m = 20; n = 10; p = 4;
A = randn(m,n); b = randn(m,1);
C = randn(p,n); d = randn(p,1); e = rand;
cvx_begin
    variable x(n);
    minimize( norm( A * x - b, 2 ) )
    subject to
        C * x == d;
        norm( x, Inf ) <= e;
cvx_end
```

$$\begin{array}{rl} \min & \|Ax - b\| \\ Cx = d & \\ \|x\|_\infty \leq e & \end{array}$$

CVX

SDP Example

```
n = 10;
E1 = randn(n,n); d1 = randn(n,1);
E2 = randn(n,n); d2 = randn(n,1);
epsx = 1e-1;
cvx_begin sdp

    variable X(n,n) semidefinite;
    minimize trace(X);
    subject to
    X*ones(n,1) == zeros(n,1);
    abs(trace(E1*X)-d1)<=epsx;
    abs(trace(E2*X)-d2)<=epsx;

cvx_end
```

$$\begin{array}{ll} \text{minimize} & trace(X) \\ \text{subject to} & X = X^T \geq 0 \\ & X \cdot 1 = 0 \\ & |trace(E_1 X) - d_1| \leq \varepsilon \\ & |trace(E_2 X) - d_2| \leq \varepsilon \end{array}$$

References

- S. Boyd, L. Vandenberghe, **Convex Optimization**, available online at: http://stanford.edu/ boyd/cvxbook/
- F. Chung, **Spectral Graph Theory**, AMS 1997.
- [3]A. Javanmard, A. Montanari, Localization from Incomplete Noisy Distance Measurements, arXiv:1103.1417, Nov. 2012; also ISIT 2011.