

Least-Squares Intersection of Lines

Johannes Traa - UIUC 2013

This write-up derives the least-squares solution for the intersection of lines. In the general case, a set of lines will not intersect at a single point. A least-squares solution can be derived that minimizes the sum of perpendicular distances from the unique solution point to all the lines. L2 regularization is also considered to bias the solution towards a reference point. The solution for the case when only the direction vector parameters of the lines are known is also derived. This involves solving for the null space eigenvector of a matrix.

1 PARAMETERIZATION

Consider a line in 2 dimensions. It can be fully described by a point on the line:

$$\mathbf{a} = \begin{bmatrix} a_1 & a_2 \end{bmatrix}^T, \quad (1)$$

and a direction vector:

$$\mathbf{n} = \begin{bmatrix} n_1 & n_2 \end{bmatrix}^T, \quad \|\mathbf{n}\|_2 = \mathbf{n}^T \mathbf{n} = 1. \quad (2)$$

Thus, the line is of the form:

$$\mathbf{p} = \mathbf{a} + t \mathbf{n}, \quad -\infty < t < \infty. \quad (3)$$

2 DISTANCE FROM A POINT TO A LINE

The squared perpendicular distance from a point to a line can be derived from the geometry involved (see Figure 1):

$$D(\mathbf{p}; \mathbf{a}, \mathbf{n}) = \|(\mathbf{a} - \mathbf{p}) - ((\mathbf{a} - \mathbf{p})^T \mathbf{n}) \mathbf{n}\|_2^2 \quad (4)$$

$$= \|(\mathbf{a} - \mathbf{p}) - \mathbf{n} \mathbf{n}^T (\mathbf{a} - \mathbf{p})\|_2^2 \quad (5)$$

$$= \|(\mathbf{I} - \mathbf{n} \mathbf{n}^T) (\mathbf{a} - \mathbf{p})\|_2^2 \quad (6)$$

$$= (\mathbf{a} - \mathbf{p})^T (\mathbf{I} - \mathbf{n} \mathbf{n}^T) (\mathbf{I} - \mathbf{n} \mathbf{n}^T)^T (\mathbf{a} - \mathbf{p}) \quad (7)$$

$$= (\mathbf{a} - \mathbf{p})^T (\mathbf{I} - \mathbf{n} \mathbf{n}^T) (\mathbf{a} - \mathbf{p}), \quad (8)$$

The last line follows since the matrix $\mathbf{I} - \mathbf{n} \mathbf{n}^T$ is a projector (onto the orthocomplement of \mathbf{n}) and is thus idempotent. It removes from a vector its component in the direction of \mathbf{n} . This operation will only cause a change the first time it is applied. Mathematically:

$$(\mathbf{I} - \mathbf{n} \mathbf{n}^T) (\mathbf{I} - \mathbf{n} \mathbf{n}^T)^T = \mathbf{I} - 2\mathbf{n} \mathbf{n}^T + \mathbf{n} \mathbf{n}^T \mathbf{n} \mathbf{n}^T = \mathbf{I} - \mathbf{n} \mathbf{n}^T. \quad (9)$$

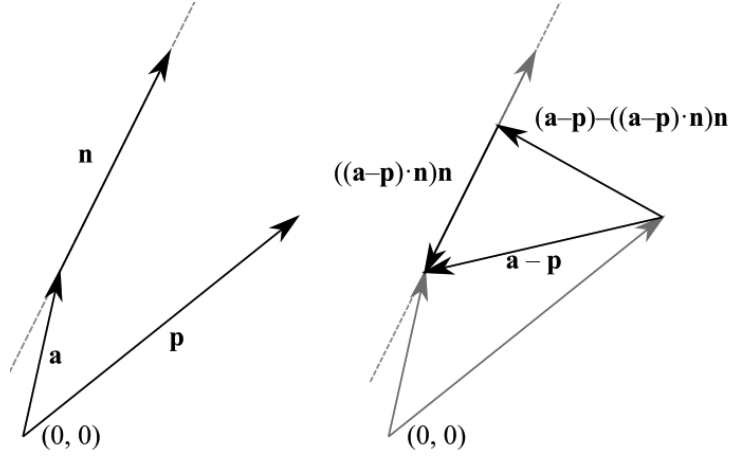


Figure 1: Perpendicular distance from a point to a line. (Source: Wikipedia)

2.1 INTERPRETATION OF COST FUNCTION AS GAUSSIAN ERROR

We can interpret the quadratic distance function as corresponding to a Gaussian error term. The matrix $\mathbf{I} - \mathbf{n}\mathbf{n}^T$ serves to project the vectors \mathbf{a} and \mathbf{p} into the space orthogonal to \mathbf{n} . Thus, minimizing the distance to the line in this projected space corresponds to maximizing the pdf of a Gaussian distribution.

3 LEAST-SQUARES SOLUTION TO THE INTERSECTION OF LINES

If we have K lines, they may all intersect at a unique point, only in pairs, or in other ways. We can find the unique solution to the “intersection” of the set of lines in a least-squares sense. The idea is to minimize the sum of squared distances:

$$D(\mathbf{p}; \mathbf{A}, \mathbf{N}) = \sum_{j=1}^K D(\mathbf{p}; \mathbf{a}_j, \mathbf{n}_j) = \sum_{j=1}^K (\mathbf{a}_j - \mathbf{p})^T (\mathbf{I} - \mathbf{n}_j \mathbf{n}_j^T) (\mathbf{a}_j - \mathbf{p}) \quad . \quad (10)$$

This cost function corresponds to a product of independent Gaussian pdf's. Thus, the objective is:

$$\hat{\mathbf{p}} = \underset{\mathbf{p}}{\operatorname{argmin}} D(\mathbf{p}; \mathbf{A}, \mathbf{N}) \quad , \quad (11)$$

which is quadratic in \mathbf{p} . Taking derivatives with respect to \mathbf{p} , we have:

$$\frac{\partial D}{\partial \mathbf{p}} = \sum_{j=1}^K -2 (\mathbf{I} - \mathbf{n}_j \mathbf{n}_j^T) (\mathbf{a}_j - \mathbf{p}) = 0 \quad . \quad (12)$$

Re-arranging, we get a linear system of equations:

$$\mathbf{R}\mathbf{p} = \mathbf{q} \quad , \quad (13)$$

$$\mathbf{R} = \sum_{j=1}^K (\mathbf{I} - \mathbf{n}_j \mathbf{n}_j^T) \quad , \quad \mathbf{q} = \sum_{j=1}^K (\mathbf{I} - \mathbf{n}_j \mathbf{n}_j^T) \mathbf{a}_j \quad , \quad (14)$$

We can either solve the system directly or apply the Moore-Penrose pseudoinverse:

$$\hat{\mathbf{p}} = \mathbf{R}^\dagger \mathbf{q} \quad . \quad (15)$$

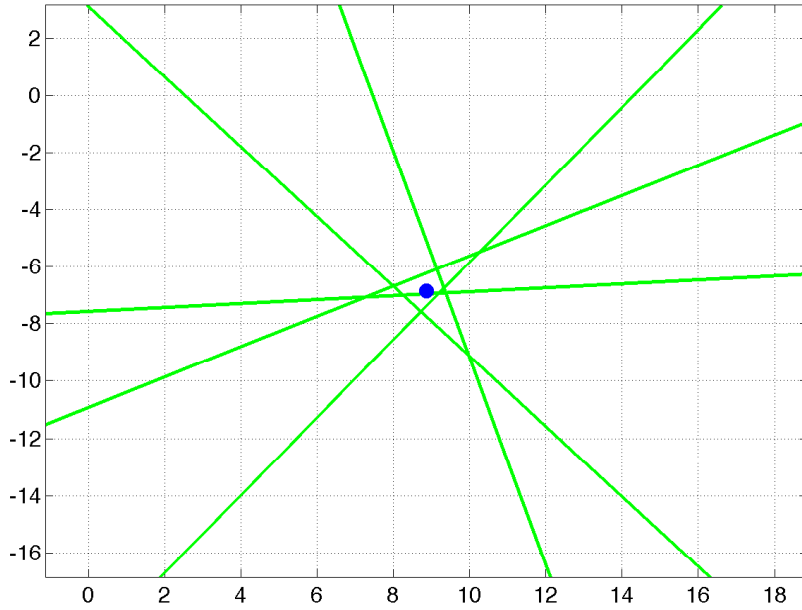


Figure 2: (Top) Least-squares solution to intersection of $K = 5$ lines in 2 dimensions.

This generalizes to any number of dimensions. An example for $K = 5$ lines in 2 dimensions is shown in Figures 2 and 3. The latter plot shows that the objective is quadratic and that the optimum is achieved at the least-squares solution.

3.1 LINE CONFIDENCE

We might want to incorporate a confidence value, c_j , that captures how severe it is to be far away from each line. In that case, we can minimize an alternative cost function:

$$D(\mathbf{p}; \mathbf{A}, \mathbf{N}, \mathbf{c}) = \sum_{j=1}^K c_j (\mathbf{a}_j - \mathbf{p})^T (\mathbf{I} - \mathbf{n}_j \mathbf{n}_j^T) (\mathbf{a}_j - \mathbf{p}) , \quad (16)$$

which leads once again to a system of linear equations:

$$\mathbf{R}\mathbf{p} = \mathbf{q} , \quad (17)$$

$$\mathbf{R} = \sum_{j=1}^K c_j (\mathbf{I} - \mathbf{n}_j \mathbf{n}_j^T) , \quad \mathbf{q} = \sum_{j=1}^K c_j (\mathbf{I} - \mathbf{n}_j \mathbf{n}_j^T) \mathbf{a}_j . \quad (18)$$

In the probabilistic interpretation, the confidence scores are precisions (inverse variances) of the Gaussian distributions. If the confidence for a particular line is relatively very large, then the solution will lie very near that line.

3.2 BIAS TOWARDS A REFERENCE POINT

We may want the solution to be reasonably close to a reference point \mathbf{s} . This might be useful in an adaptive algorithm that updates the “intersection” over time. We can enforce this by regularizing the solution with a quadratic term:

$$D(\mathbf{p}; \mathbf{A}, \mathbf{N}, \mathbf{s}, \lambda) = \sum_{j=1}^K (\mathbf{a}_j - \mathbf{p})^T (\mathbf{I} - \mathbf{n}_j \mathbf{n}_j^T) (\mathbf{a}_j - \mathbf{p}) + \lambda (\mathbf{s} - \mathbf{p})^T (\mathbf{s} - \mathbf{p}) . \quad (19)$$

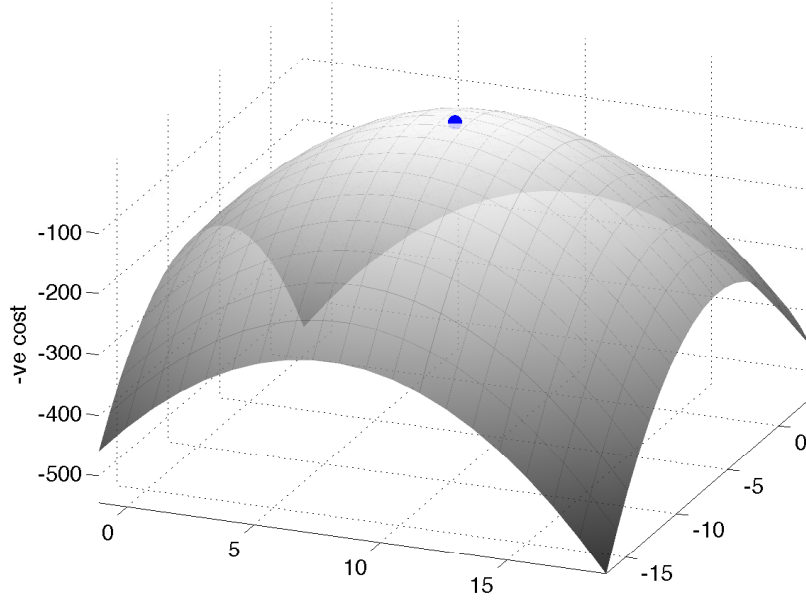


Figure 3: Negative of quadratic objective function, $-D(\mathbf{p}; \mathbf{A}, \mathbf{N})$, showing that the optimum is achieved.

Following the derivation from before, we get the following system of linear equations:

$$\mathbf{R}\mathbf{p} = \mathbf{q} \quad , \quad (20)$$

$$\mathbf{R} = \sum_{j=1}^K (\mathbf{I} - \mathbf{n}_j \mathbf{n}_j^T) + \lambda \mathbf{I} \quad , \quad \mathbf{q} = \sum_{j=1}^K (\mathbf{I} - \mathbf{n}_j \mathbf{n}_j^T) \mathbf{a}_j + \lambda \mathbf{s} \quad . \quad (21)$$

From this, we can see that as $\lambda \rightarrow \infty$, $\mathbf{p} \rightarrow \mathbf{s}$.

4 NULL SPACE SOLUTION FOR PARTIALLY KNOWN LINE PARAMETERS

We can derive an eigendecomposition solution for the case where only the direction vectors \mathbf{n}_j are known. For a solution to exist, we need at least 3 lines and 3 sets of direction vectors (3 for each line). The cost function then becomes:

$$D(\mathbf{P}, \mathbf{A}; \mathbf{N}) = \sum_{i=1}^L \sum_{j=1}^K (\mathbf{a}_j - \mathbf{p}_i)^T (\mathbf{I} - \mathbf{n}_{ij} \mathbf{n}_{ij}^T) (\mathbf{a}_j - \mathbf{p}_i) \quad , \quad (22)$$

which we would like to minimize over all choices of:

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}_1 & \cdots & \mathbf{p}_L \end{bmatrix} \quad , \quad \mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \cdots & \mathbf{a}_K \end{bmatrix} \quad . \quad (23)$$

For simplicity, let's consider the minimal case where $L = K = 3$. If we stack all the parameters to be learned in a single column vector:

$$\mathbf{x} = \begin{bmatrix} \mathbf{a}_1^T & \mathbf{a}_2^T & \mathbf{a}_3^T & \mathbf{p}_1^T & \mathbf{p}_2^T & \mathbf{p}_3^T \end{bmatrix}^T \quad , \quad (24)$$

then Equation 22 can be written as a single quadratic term in \mathbf{x} :

$$D(\mathbf{x}; \mathbf{N}) = \mathbf{x}^T \mathbf{C} \mathbf{x} \quad , \quad (25)$$

where

$$\mathbf{C} = \mathbf{G}^T \mathbf{B} \mathbf{G} \text{ ,} \quad (26)$$

and

$$\mathbf{G} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & -\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & -\mathbf{I} \end{bmatrix} \text{ ,} \quad (27)$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{21} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{M}_{31} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{M}_{12} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{M}_{22} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{M}_{32} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{M}_{13} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{M}_{23} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{M}_{33} \end{bmatrix} \text{ ,} \quad \mathbf{M}_{ij} = \mathbf{I} - \mathbf{n}_{ij} \mathbf{n}_{ij}^T \text{ .} \quad (28)$$

This can be derived by stacking all the difference vectors:

$$\mathbf{q} = \begin{bmatrix} \mathbf{a}_1 - \mathbf{p}_1 \\ \mathbf{a}_1 - \mathbf{p}_2 \\ \mathbf{a}_1 - \mathbf{p}_3 \\ \mathbf{a}_2 - \mathbf{p}_1 \\ \mathbf{a}_2 - \mathbf{p}_2 \\ \mathbf{a}_2 - \mathbf{p}_3 \\ \mathbf{a}_3 - \mathbf{p}_1 \\ \mathbf{a}_3 - \mathbf{p}_2 \\ \mathbf{a}_3 - \mathbf{p}_3 \end{bmatrix} \text{ ,} \quad (29)$$

and determining what matrix \mathbf{G} is necessary to construct \mathbf{q} from \mathbf{x} , i.e. $\mathbf{q} = \mathbf{G}\mathbf{x}$. Then, we can write the cost function in the form:

$$\mathbf{q}^T \mathbf{B} \mathbf{q} = (\mathbf{G}\mathbf{x})^T \mathbf{B} (\mathbf{G}\mathbf{x}) = \mathbf{x}^T (\mathbf{G}^T \mathbf{B} \mathbf{G}) \mathbf{x} = \mathbf{x}^T \mathbf{C} \mathbf{x} \text{ .} \quad (30)$$

The matrix \mathbf{C} turns out to be:

$$\mathbf{C} = \begin{bmatrix} \mathbf{M}_{11} + \mathbf{M}_{21} + \mathbf{M}_{31} & \mathbf{0} & \mathbf{0} & -\mathbf{M}_{11} & -\mathbf{M}_{21} & -\mathbf{M}_{31} \\ \mathbf{0} & \mathbf{M}_{12} + \mathbf{M}_{22} + \mathbf{M}_{32} & \mathbf{0} & -\mathbf{M}_{12} & -\mathbf{M}_{22} & -\mathbf{M}_{32} \\ \mathbf{0} & \mathbf{0} & \mathbf{M}_{13} + \mathbf{M}_{23} + \mathbf{M}_{33} & -\mathbf{M}_{13} & -\mathbf{M}_{23} & -\mathbf{M}_{33} \\ -\mathbf{M}_{11} & -\mathbf{M}_{12} & -\mathbf{M}_{13} & \mathbf{M}_{11} + \mathbf{M}_{12} + \mathbf{M}_{13} & \mathbf{0} & \mathbf{0} \\ -\mathbf{M}_{21} & -\mathbf{M}_{22} & -\mathbf{M}_{23} & \mathbf{0} & \mathbf{M}_{21} + \mathbf{M}_{22} + \mathbf{M}_{23} & \mathbf{0} \\ -\mathbf{M}_{31} & -\mathbf{M}_{32} & -\mathbf{M}_{33} & \mathbf{0} & \mathbf{0} & \mathbf{M}_{31} + \mathbf{M}_{32} + \mathbf{M}_{33} \end{bmatrix} \quad (31)$$

Minimizing Equation (30) as is yields the trivial solution $\mathbf{x} = \mathbf{0}$. There are also unresolved translation, scale, and reflection ambiguities that make \mathbf{C} rank deficient by 3 (for the 2D case). So, we will impose a constraint to make the solution unique up to a scaling factor:

$$\mathbf{x}^T \mathbf{H} \mathbf{x} = 0 \quad , \quad (32)$$

where

$$\mathbf{H} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad . \quad (33)$$

This removes the translation ambiguity by forcing $\mathbf{a}_1 = \mathbf{0}$, which in turn removes two degrees of freedom from the solution. Now, the problem is to minimize:

$$D(\mathbf{x}; \mathbf{N}) = \mathbf{x}^T \mathbf{C} \mathbf{x} + \mathbf{x}^T \mathbf{H} \mathbf{x} \quad . \quad (34)$$

Taking derivatives with respect to \mathbf{x} and setting the result equal to zero, we get the homogeneous system of equations:

$$(\mathbf{C} + \mathbf{H}) \mathbf{x} = \mathbf{0} \quad . \quad (35)$$

The solution for \mathbf{x} is the eigenvector \mathbf{v} that spans the null space of $\mathbf{C} + \mathbf{H}$. Its corresponding eigenvalue is $\lambda = 0$. The vector $\hat{\mathbf{x}} = \mathbf{v}$ minimizes the original cost function in Equation (22) and provides a unique solution up to a real-valued scaling factor. The reflection about \mathbf{a}_1 (i.e. the origin) is resolved by ensuring that the direction vectors implied by the solution are oriented correctly:

$$\hat{\mathbf{x}} = \begin{cases} \mathbf{v} & , \quad (\mathbf{p}_1 - \mathbf{a}_1)^T \mathbf{n}_{11} > 0 \\ -\mathbf{v} & , \quad (\mathbf{p}_1 - \mathbf{a}_1)^T \mathbf{n}_{11} < 0 \end{cases} \quad . \quad (36)$$

Thus, we can recover both the lines *and* the intersection points up to a positive scaling factor from just the 9 direction vectors \mathbf{n}_{ij} .

4.1 HANDLING ERRORS IN THE DIRECTION VECTORS

When the direction vectors are not exact, multiple eigenvectors may have very small eigenvalues. This might happen if the \mathbf{a} vectors are collinear and/or the \mathbf{p} vectors are collinear. Numerically speaking, this makes the solution ill-conditioned. In this case, we can regularize the solution to be robust to such uncertainty by adding a scaled matrix of ones to the system:

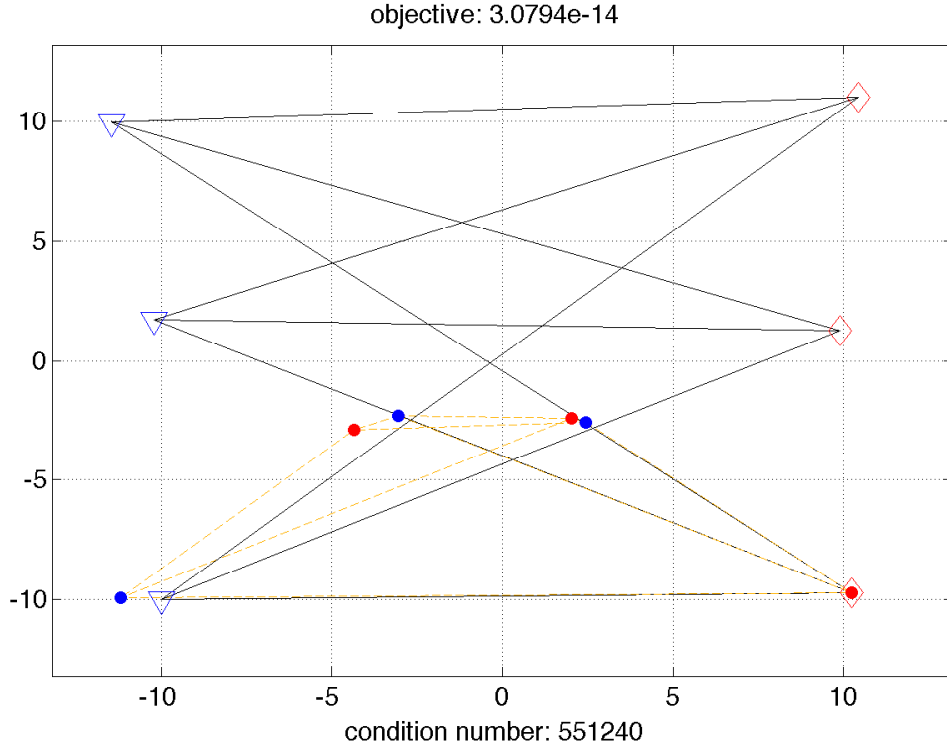


Figure 4: Poorly conditioned problem. The true and estimated solutions are denoted by the empty and filled symbols, respectively. Because of the geometry involved, the solution is very sensitive to small Gaussian errors ($\sigma = 0.1$) in the direction vectors. The condition number is measured as the ratio of the largest eigenvalue of $\mathbf{C} + \mathbf{H}$ to its second smallest.

$$(\mathbf{C} + \mathbf{H} + \lambda \mathbf{J}) \mathbf{x} = \mathbf{0} \quad , \quad \mathbf{J} = \mathbf{11}^T \quad . \quad (37)$$

An example is shown in Figures 4 and 5. Although the regularization makes the value of the cost function non-zero, it allows a reasonable solution to be found.

5 LEAST-SQUARES SOLUTION FOR UNKNOWN LINE ORIENTATIONS

Consider the case where the line position parameters \mathbf{p}_j are known but the direction vectors \mathbf{n}_{ij} are only known up to a rotation angle θ_j . This means that the relative orientation of the direction vectors is only known for each line, but not globally for all the lines. We can represent this ambiguity via a rotation matrix for each line:

$$\mathbf{R}(\theta_j) = \begin{bmatrix} \cos(\theta_j) & -\sin(\theta_j) \\ \sin(\theta_j) & \cos(\theta_j) \end{bmatrix} \quad . \quad (38)$$

Then, the cost function to minimize is:

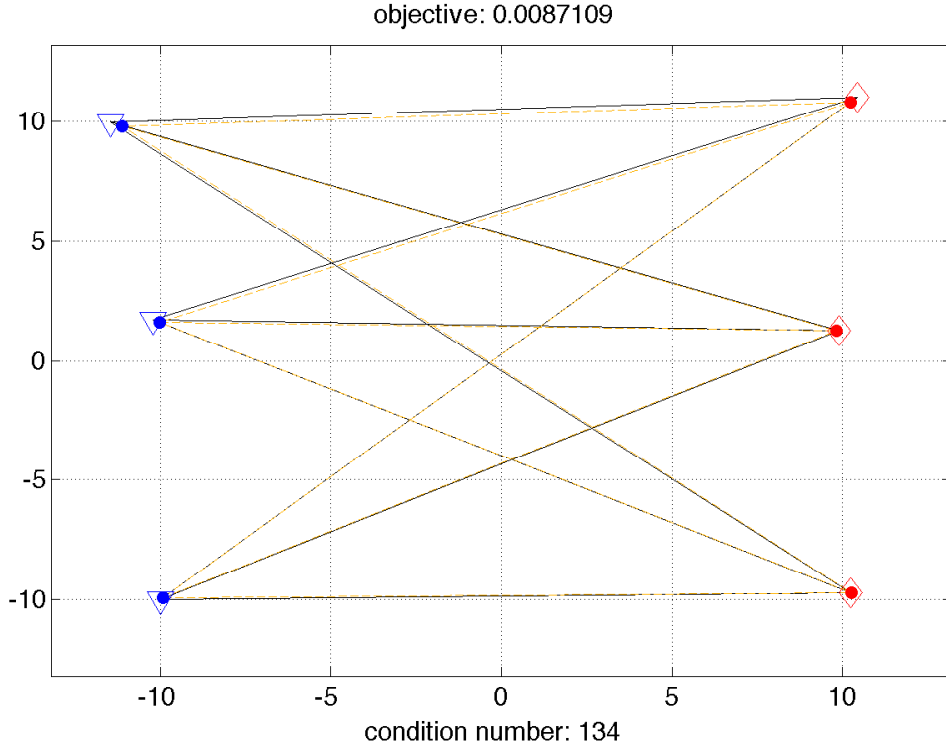


Figure 5: Regularized solution with $\lambda = 0.3$. The objective function value is non-zero, but the solution is more reasonable.

$$D(\mathbf{P}, \boldsymbol{\theta}; \mathbf{N}, \mathbf{A}) = \sum_{i=1}^L \sum_{j=1}^K (\mathbf{a}_j - \mathbf{p}_i)^T \left(\mathbf{I} - \mathbf{R}(\theta_j) \mathbf{n}_{ij} \mathbf{n}_{ij}^T \mathbf{R}(\theta_j)^T \right) (\mathbf{a}_j - \mathbf{p}_i) \quad (39)$$

$$= \sum_{i=1}^L \sum_{j=1}^K (\mathbf{a}_j - \mathbf{p}_i)^T \mathbf{R}(\theta_j) (\mathbf{I} - \mathbf{n}_{ij} \mathbf{n}_{ij}^T) \mathbf{R}(\theta_j)^T (\mathbf{a}_j - \mathbf{p}_i) \quad (40)$$

$$= \sum_{i=1}^L \sum_{j=1}^K (\mathbf{a}_j - \mathbf{p}_i)^T \mathbf{R}(\theta_j) \mathbf{M}_{ij} \mathbf{R}(\theta_j)^T (\mathbf{a}_j - \mathbf{p}_i) \quad (41)$$

We can solve for the intersections \mathbf{p}_i and orientations θ_j in a least-squares sense with at least $K = 3$ lines and $L = 2$ intersections. The non-linear dependence on θ_j makes this a more challenging optimization problem. The circular nature of the orientation angles evokes an objective function with many local minima. The optimization must be initialized quite close to the global minimum for it to be achieved.

If the point parameters \mathbf{a}_j , orientations θ_j , and intersections \mathbf{p}_i are *all* unknown, then the cost function is the same, but is highly non-convex in the full parameter set. So, it will most likely be very difficult to find the solution or even to determine if a reasonable solution exists in the case of errors in the only known information: the direction vectors \mathbf{n}_{ij} .