16-811

Erica Weng

September 2020

3

a

there are $\boxed{infinitely}$ many exact solutions, given by the function below.

$$V : \begin{pmatrix} -0.74 & 0.13 & 0.66 \\ -0.10 & -0.99 & 0.09 \\ -0.66 & -0.01 & -0.75 \end{pmatrix} \frac{1}{\Sigma} : \begin{pmatrix} 0.06 & 0.00 & 0.00 \\ 0.00 & 0.66 & 0.00 \\ 0.00 & 0.00 & 0.00 \end{pmatrix} U^T : \begin{pmatrix} -0.09 & -0.79 & -0.61 \\ -0.57 & -0.46 & 0.68 \\ -0.82 & 0.41 & -0.41 \end{pmatrix}$$

$$x = V\frac{1}{\Sigma}U^T b = \boxed{\begin{pmatrix} 0.02 \\ 0.86 \\ 0.12 \end{pmatrix}}$$

verify:

$$Ax - b = \begin{pmatrix} 0.00 \\ -0.00 \\ 0.00 \end{pmatrix}$$

since there are infinitely many solutions, the solutions are given by

$$\boxed{c \cdot \begin{pmatrix} 0.02 \\ 0.86 \\ 0.12 \end{pmatrix}}$$

for all value of c.

b

SVD same as in (a).
there $\boxed{0}$ exact solutions, the "least mean-squares" solution is given below

$$x = V\frac{1}{\Sigma}U^T b = \boxed{\begin{pmatrix} 0.02 \\ 0.86 \\ 0.12 \end{pmatrix}}$$

verify:

$$Ax - b = \begin{pmatrix} -2.00 \\ 1.00 \\ -1.00 \end{pmatrix}$$

which is orthogonal to A's column space (spanned by $\begin{pmatrix} 1.00 \\ 2.00 \\ 0.00 \end{pmatrix}$ and $\begin{pmatrix} 1.00 \\ 9.00 \\ 7.00 \end{pmatrix}$).

for parts (a) and (b) is that the SVD decomposition is the same. However, part (a)'s b is in A's column space, so $Ax - b = 0$, whereas in part (b) b is not in A's column space, so $Ax - b$ is a vector perpendicular to A's column space.

c

The code used to generate these solutions is in `code/q3.py`
there is $\boxed{1}$ exact solution.

$$V : \begin{pmatrix} -0.69 & 0.32 & -0.65 \\ 0.73 & 0.31 & -0.61 \\ 0.01 & -0.89 & -0.45 \end{pmatrix} \frac{1}{\Sigma} : \begin{pmatrix} 0.07 & 0.00 & 0.00 \\ 0.00 & 0.17 & 0.00 \\ 0.00 & 0.00 & 0.54 \end{pmatrix} U^T : \begin{pmatrix} -0.98 & -0.20 & -0.10 \\ 0.02 & -0.51 & 0.86 \\ -0.22 & 0.84 & 0.50 \end{pmatrix}$$

$$x = V\frac{1}{\Sigma}U^T b = \boxed{\begin{pmatrix} -1.00 \\ -2.00 \\ -3.00 \end{pmatrix}}$$

5

The problem calls us to find the best rotation matrix R and translation t for the two sets of points $p_1, \ldots, p_n$ and $q_1, \ldots, q_n$, where $q_i = Rp_i + t$. That means solving the least-squares solution to the equation

$$L = \arg\min_{R,t} \sum_{i=1}^{n} ||Rp_i + t - q_i||^2$$

.

Solving for t is simple:

$$\frac{\partial L}{\partial t} = 2\Big(R\sum_{i=1}^{n} p_i + nt - \sum_{i=1}^{n} q_i\Big) \quad = 0$$
$$t = \bar{q} - R\bar{p}$$

Solving for R now:

$$L = \arg\min_{R} \sum_{i=1}^{n} ||Rp_i + (\bar{q} - R\bar{p}) - q_i||^2$$
$$L = \arg\min_{R} \sum_{i=1}^{n} ||R(p_i - \bar{p}) - (q_i - \bar{q})||^2$$

change variables for ease of calculation using $p' = p_i - \bar{p}$ and $q' = q_i - \bar{q}$

$$
\begin{aligned}
L &= \arg\min_R \sum_{i=1}^{n} ||Rp'_i - q'_i||^2 \\
&= \arg\min_R \sum_{i=1}^{n} (Rp'_i - q'_i)^T (Rp'_i - q'_i) \\
&= \arg\min_R \sum_{i=1}^{n} (Rp'_i - q'_i)^T (Rp'_i - q'_i) \\
&= \arg\min_R \sum_{i=1}^{n} p_i'^T R^T R p'_i - q_i'^T R p'_i - p_i'^T R^T q'_i + q_i'^T q'_i \\
&= \arg\min_R \sum_{i=1}^{n} p_i'^T p'_i - 2 p_i'^T R^T q'_i + q_i'^T q'_i \\
&= \arg\min_R \sum_{i=1}^{n} -Tr(R p'_i q_i'^T) \\
&= \arg\max_R Tr(R P' Q'^T)
\end{aligned}
$$

the third-to-last line gotten by $R^T R = I$ because rotation matrices are orthogonal, and because $q_i'^T R p_i$ is a scalar and $q_i'^T R p_i = (p'_i R^T q'_i)^T$, thus $q_i'^T R p_i = p'_i R^T q'_i$. Then the second-to-last line gotten by taking out constants and factors that don't depend on R, and using Hint #2. The last line gotten by putting all column-vector points p_i into a matrix $P = [p_1; \ldots; p_n]$ and similarly for all q_i into matrix $Q = [q_1, \ldots; q_n]$. Now, computing the SVD of $P'Q'^T$:

$$
\begin{aligned}
&= \arg\max_R Tr(RU\Sigma V^T) \\
&= \arg\max_R Tr(\Sigma V^T R U)
\end{aligned}
$$

because $Tr(AB) = Tr(BA)$. because V, U, and R are all orthogonal, and any multiplication of them will give another orthogonal matrix, then to maximize $Tr(\Sigma V^T R U)$ means to have $V^T R U = I$ where $Tr(\Sigma V^T R U) = Tr(\Sigma)$. So $R = V U^T$.

So now here is the algorithm in the file `code/q5.py`:

1. For sets of points $P = [p_i; \ldots; p_n]$ and $Q = [q_i; \ldots; q_n]$, find the centroid (average value for each of the 3 dimensions) of each set of points, $\bar{p} = \frac{1}{n}\sum_{i=1}^{n} p_i$ and $\bar{q} = \frac{1}{n}\sum_{i=1}^{n} q_i$, respectively

2. Find new sets of points $P' = P - \bar{p}$ and $Q' = Q - \bar{q}$ to get the translation from each set of points to the origin

3. Use the SVD to solve for the $P'Q'^T = U\Sigma V^T$, and get rotation matrix $R = VU^T$

4. Calculate translation $t = \bar{q} - R\bar{p}$
5. return R and t.

To run the implementation, run `python3 code/q5.py`. It will run several several test cases.