Jindong_Shi
2021-09-26

第二节课习题

2. 熟悉 Eigen 矩阵运算

2.1 在什么条件下，x 有解且唯一

在 A 为方阵的前提下，当 A 满秩的时候，Ax = b 中 x 有唯一解

2.2 高斯消元法的原理是什么?

利用有限次初等变换将增广矩阵 [A|b] 转化为行阶梯阵(上三角矩阵)，然后通过回代过程依次解出 x_n，x_{n-1}, …, x_2, x_1.

2.3 QR 分解的原理是什么?

QR 分解将矩阵分解为 Q*R 的乘积形式，其中 Q 为正交矩阵($Q^TQ = I$)，R 为上三角矩阵。QR 分解经常用来解线性最小二乘法问题。QR 分解也是特定特征值算法即 QR 算法的基础。QR 分解的实际计算有很多方法，例如 Givens 旋转、Householder 变换，以及 Gram-Schmidt 正交化等等。

2.4 Cholesky 分解的原理是什么?

Cholesky 分解是将一个对称正定矩阵分解为 $L*L^T$ 的乘积形式，其中 L 为下三角矩阵。由于 Cholesky 分解的前提是对称正定矩阵，所以矩阵的所以特征值都是大于零的，导致分解的下三角矩阵的对角线元素也是大于零的。求解 Ax = b 时，可利用 Cholesky 分解，将等式转化为 $LL^Tx = b$，然后利用前向迭代求解 Ld = b 中的 d, 再利用后向迭代解出 $L^Tx = d$ 中 x。

2.5 编程实现 A 为 100 x 100 随机矩阵时，用 QR 和 Cholesky 分解求 x 的程序，可以参考本次课程用到的 useEigen 例程。

程序源文件 qr_cholesky_test.cpp 和可执行文件 qr_cholesky_test 位于文件夹 QR_Cholesky_Test 内。

3. 矩阵论基础

3.1 什么是正定矩阵和半正定矩阵?

1) 设 A 是 n 阶方阵，如果对任何非零向量 X，都有 $X^TAX > 0$，其中 X^T 表示 X 的转置，则 A 为正定矩阵。
对于复数的情况，如果对任何非零的复向量 X，都有 $X^HAX > 0$, 其中 X^H 表示 X 的共轭转置，则 A 为正定矩阵。

2) 设 A 是 n 阶方阵，如果对任何非零向量 X，都有 $X^TAX \geq 0$，其中 X^T 表示 X 的转置，则 A 为半正定矩阵。
对于复数的情况，如果对任何非零的复向量 X，都有 $X^HAX \geq 0$, 其中 X^H 表示 X 的共轭转置，则 A 为半正定矩阵。

性质 1: 若 A 为正定矩阵，则 A 的特征值均为正。若 A 为半正定矩阵，则 A 的特征值均为非负。
性质 2: 若矩阵A满足$A = B^TB$，则该对称矩阵A一定至少是一个半正定矩阵，因为对于任何非零向量X，都有$X^TAX = X^TB^TBX = \|BX\|^2 \geq 0$; 当 $Bx = 0$ 不存在非零解时，即B列满秩时，则A为正定矩阵。

3.2 对于方阵 A，它的特征值是什么? 特征向量是什么? 特征值一定是实数吗? 如何计算一个矩阵的特征值?

1) 设 A 是 n 阶方阵，如果存在数 λ 和非零 n 维列向量 x，使得 $Ax = \lambda x$ 成立，则称 λ 是 A 的一个特征值，非零 n 维列向量x称为矩阵 A 对应于特征值 λ 的特征向量。
2) 特征值不一定是实数，也可能是复数(包括纯虚数)。
3) 计算矩阵特征值的步骤:
    1. 计算满足行列式$det(A - \lambda I) = 0$ 的所有 λ 值即为 A 的所有特征值。
    2. 对于每一个特征值 λ 求出对应的非零特征向量x，满足$(A - \lambda I)x = 0$

3.3 什么是矩阵的相似性? 相似性反映了什么几何意义?

1) 相似矩阵指存在相似关系的矩阵。两个$n \times n$矩阵A与B为相似矩阵当且仅当存在一个$n \times n$的可逆矩阵P，使得:

$$P^{-1}AP = B$$

P被称为矩阵A与B之间的相似变换矩阵。判断两个矩阵是否相似的必要而非充分条件为:
1.判断特征值是否相等；2.判断行列式是否相等；3.判断迹是否相等；4.判断秩是否相等。
2) 相似性的几何意义: 相似的矩阵是同一个线性变换在不同基/坐标系下的的不同描述。 线性变换若粗略看成一个刚体的特定运动。 刚体的特定运动是同一个，但坐标系改变的话这个运动的描述函数就会不一样，如果这个函数可用矩阵等价替代的话，一个坐标系就对应着一个矩阵，因此这些矩阵就不同，但这些矩阵必有关系，这个关系就是相似。
直观一点描述:
若 A 是线性变换 L 在基底 {a_1, ···, a_n} 下的表示矩阵
B 是线性变换 L 在基底 {b_1, ···, bn} 下的表示矩阵
那么相似变换矩阵 P 就是 从 {a_1, ···, a_n} 到 {b_1, ···, bn} 的线性变换 在 {a_1, ···, a_n} 坐标系下的表示矩阵。

3.4 矩阵一定能对角化吗? 什么样的矩阵能保证对角化? 不能对角化的矩阵能够形成什么样的形式(Jordan 标准形)?

1) 矩阵不一定能对角化。若S为特征向量矩阵: $S = [x_1,\ x_2, \dots,\ x_n]$, 其中$x_1,\ x_2, \dots,\ x_n$ 为矩阵A的 n 个特征向量，矩阵对角化公式为$S^{-1}AS = \Lambda$, 其中Λ为对角化矩阵，对角元素为 n 个特征值:

$$\Lambda = diag(\lambda_1, \lambda_2, \dots, \lambda_n)$$

由此可见矩阵 A 若要对角化成功，那么特征向量矩阵 S必须可逆，即S必须满秩，也就是说 A 的 n 个特征向量必须线性无关。

注意: 若矩阵A的特征值重复，并不代表矩阵A的特征向量重复，并代表其不能被对角化，例如单位矩阵 I 拥有重复的特征值，但是每个特征向量都线性无关, I可以对角化, I的对角化矩阵就是自己。

2) 如果矩阵 A 的各个特征向量线性无关，则矩阵 A 可以被对角化。

矩阵 A 的各个特征向量线性无关的一种特殊情况是实对称矩阵:

实对称矩阵能够保证对角化，因为实对称矩阵的不同特征值对应的特征向量是两两垂直(正交)的，保证了各个特征向量的线性无关性。

3) 所有的 n 阶方阵都可以化为“若尔当 Jordan 标准型”, 其中可对角化的矩阵可以化为若尔当标准型的特例----对角标准型，而不可对角化的矩阵可以化为一般性的若尔当标准型，即若干个主对角线为特征值(其中一些特征值可以相等)，下方(或上方)次对角线全为 1，其余全为 0 的若尔当块组成的矩阵。

3.5 奇异值分解 (SVD)是什么意思?

对于任意一个 mxn 阶的矩阵 A 存在一个分解，使得

$$A = U\Sigma V^T$$

其中:

U是 mxm 阶的正交矩阵，U的列向量由 AA^T 的特征向量构成。

Σ为 mxn 阶半正定对角矩阵(也就是说对角元素可能为零)，对角线上的元素称为奇异值，其中非零奇异值是 AA^T 或 A^TA 的非零特征值的平方根。

V为 nxn 阶的正交矩阵，V的列向量由 A^TA 的特征向量构成。

这样的分解就称为奇异值分解。

3.6 矩阵的伪逆是什么意思 (Psudo inverse)? 莫尔-彭多斯逆是如何定义的? 怎么计算一个矩阵的伪逆?

1) 众所周知只有非奇异的正方阵才有有逆矩阵，非正方阵或奇异正方矩阵没有逆矩阵。后来将非奇异正方阵的逆矩阵推广到非方阵以及奇异的正方阵，将得到所谓的广义逆矩阵，也称为伪逆，其具有部分逆矩阵的特性，但不一定具有逆矩阵的所有特性。对于任意一个矩阵 A，A 的伪逆 A^g 必然存在，满足 $AA^gA = A$

2) 根据莫尔-彭多斯条件可以定义不同的广义逆阵，对于任意 mxn 阶矩阵 A，莫尔-彭多斯逆 A^+需同时满足以下四个条件:

1. $A A^+A = A$
2. $A^+A A^+ = A^+$
3. $(A A^+)^H = A A^+$
4. $(A^+A)^H = A^+A$

3)

如果 A 是一个 nxn 阶方阵, 并且 A 满秩(rank(A) = n), 那么 A 的伪逆矩阵就是 A 的逆 A^{-1}

如果 A 是一个 mxn 阶矩阵, 且 m>n, 并且 A 列满秩(rank(A) = n), 那么 A 可以定义左伪逆矩阵为 $A^+ = (A^HA)^{-1}A^H$

如果 A 是一个 mxn 阶矩阵, 且 m<n, 并且 A 行满秩(rank(A) = m), 那么 A 可以定义右伪逆矩阵为 $A^+ = A^H(AA^H)^{-1}$

如果 mxn 阶矩阵 A 是任意一个秩亏损的矩阵(rank(A) < min(m, n))，那么只能通过 SVD 分解公式$A = U\Sigma V^T$来求解 A 的伪逆 A^+:

$$A^+ = V\Sigma^+U^T$$

其中Σ^+为 nxm 阶，是将Σ对角线上的非零元素取倒数后再转置得到的。

3.7 对于超定方程: Ax = b 且 A 不可逆时，我们通常计算最小二乘解: $x = arg\min_x \|Ax - b\|$。 线性方程的最小二乘解在代数意义上是可以解析地写出来地。请回答以下问题:

(a) 在 b ≠ 0 时， x 的解是什么形式? 事实上，我们可以对 A 求奇异值或对于 A^TA 求特征值。请阐述两者之间的关系。

由于$Ax = b$ 为超定方程，故A为 mxn 阶非方阵, 且 m>n，故无法通过$x = A^{-1}b$求得解。由于解的是超定方程，若A列满秩的情况下，可以使用左伪逆矩阵$A^+ = (A^TA)^{-1}A^T$。这样就得到了最小二乘解$x = A^+b = (A^TA)^{-1}A^Tb$

我们也可以通过求解矩阵A的奇异值来求解该超定方程:

先做奇异值分解$A = U\Sigma V^T$, 其中的A的非零奇异值就是AA^T 或 A^TA 的非零特征值的平方根。然后再求伪逆$A^+ = V\Sigma^+U^T$，最后可以得到解为 $x = A^+b$

Jindong_Shi
2021-09-26

(b) 当 b = 0 时，我们希望求 x 的非零解。请说明如何求解 x。

由于$Ax = 0$为超定方程，方程数量多于未知数，故不存在非零的精确解，我们只能找到一组非零的近似解x^+使得

$$x^+ = \arg\min \|Ax\|^2$$

我们可以限制$\|x\|^2 = 1$, 因为如果 x 是$Ax = 0$的一个非零解，那么kx也是一个非零解 (k为任意标量)。所以我们得到一个带约束的优化问题:

$$x^+ = \arg\min \|Ax\|^2, \text{subject to } \|x\|^2 = 1$$

利用拉格朗日法将其转化成无约束优化问题:

$$L(x,\ \lambda) = \|Ax\|^2 + \lambda(1 - \|x\|^2) = x^T A^T A x + \lambda(1 - x^T x)$$

为求极值对其进行求导:

$$\frac{\partial L(x,\ \lambda)}{\partial x} = 2A^T A x - 2\lambda x = 0$$

$$\frac{\partial L(x,\ \lambda)}{\partial \lambda} = 1 - x^T x = 0$$

由此我们得到关于最优解的等式:

$$A^T A x = \lambda x$$

我们发现 λ 和 x 分别为$A^T A$ 的特征值和特征向量，但是$A^T A$ 的特征值和特征向量有很多。我们将其带入待优化式子，得到

$$\|Ax\|^2 = x^T A^T A x = x^T \lambda x = \lambda x^T x = \lambda$$

由此可以发现$A^T A$的最小特征值 λ_{min} 就是$\|Ax\|^2$的最小值，该特征值λ_{min} 对应的特征向量 x 就是$Ax = 0$的近似解。

(c) 请谈谈你对上述解法在几何意义上的理解。该问题为开放问题。

对于求伪逆的方法解$Ax = b$的几何理解:

由于我们无法找到解x，使得 b 直接落在Ax的向量空间上，故只能求b在Ax空间上的投影，根据投影的定义有$b - Ax$垂直于空间A，故$A^T(b - Ax) = 0$, 即为$A^T A x = A^T b$, 这样就得到了最小二乘解$x = (A^T A)^{-1} A^T b$，其中A的伪逆为$A^+ = (A^T A)^{-1} A^T$。由此可见对于求解超定方程，一般适用左逆矩阵。

对于使用奇异值分解的方法解$Ax = b$的几何理解:

等式$Ax = b$结合奇异值分解$A = U\Sigma V^T$从几何意义上而言，就是利用 m x n 阶矩阵 A 将向量 x 从 V 这组正交基向量空间旋转到 U 这组正交向量基空间，并对每个方向以奇异值为缩放因子进行缩放。为了解$Ax = b$这个方程得到x, 我们可以将这个过程进行反向推导，伪逆$A^+ = V\Sigma^+ U^T$就是将 b 从 U 这组正交基向量空间旋转到 V 这组正交向量基空间，并对每个方向以奇异值的倒数为缩放因子进行反向缩放。这样就得到解 $x = A^+ b = V\Sigma^+ U^T b$

4. 几何运算练习

4.1 说明一个激光传感器下看的的点应该如何计算它的世界坐标。

激光传感器下看到的点v_L可经过三次坐标变换计算得到世界坐标，分别为

从激光传感器系到 Body 系:

通过平移加旋转 $v_B = R_{BL} v_L + t_{BL}$ 或者 通过齐次坐标的欧式坐标变换 $\begin{bmatrix} v_B \\ 1 \end{bmatrix} = T_{BL} \begin{bmatrix} v_L \\ 1 \end{bmatrix}$

再从 Body 系到 R 系:

通过平移加旋转 $v_R = R_{RB} v_B + t_{RB}$ 或者 通过齐次坐标的欧式坐标变换 $\begin{bmatrix} v_R \\ 1 \end{bmatrix} = T_{RB} \begin{bmatrix} v_B \\ 1 \end{bmatrix}$

最后再从 R 系到世界系 W:

通过平移加旋转 $v_W = R_{WR} v_R + t_{WR}$ 或者 通过齐次坐标的欧式坐标变换 $\begin{bmatrix} v_W \\ 1 \end{bmatrix} = T_{WR} \begin{bmatrix} v_R \\ 1 \end{bmatrix}$

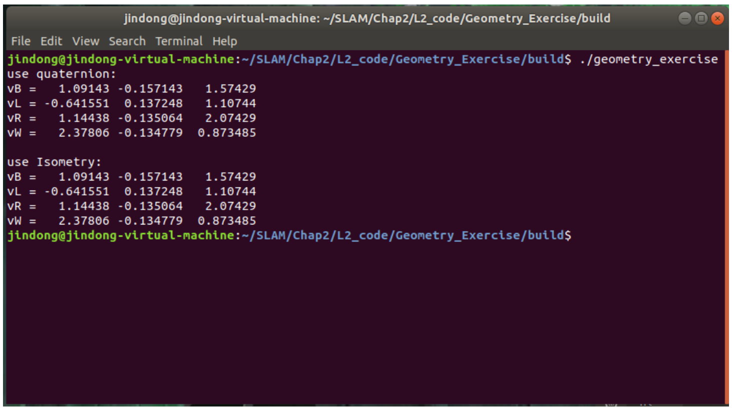
4.2 计算这个点在激光系下的坐标和世界系下的坐标

现在只知道该点在相机传感器系下的坐标为 $v_C = [0.3, 0.2, 1.2]^T$

(a) 求该点在激光系下的坐标，可以按照 相机系 C >>> Body 系 B >>> 激光系 L 进行转化

(b) 求该点在世界系下的坐标，可以按照 相机系 C >>> Body 系 B >>> 机器人系 R >>> 世界系 W 进行转化

程序源代码见文件夹 Geometry_Exercise, 程序先使用四元数(quaternion)进行坐标转换，再使用欧式变换(Isometry)对求得的结果进行验证，两种方法结果一致，见下图。



Jindong_Shi
2021-09-26

5. 旋转的表达

6. 罗德里格斯公式的证明

6.1 证明 $\boldsymbol{R = \cos\theta \cdot I + (1 - \cos\theta) n n^T + \sin\theta \cdot n}$^

证: 我们让向量 v 绕着方向 n 旋转角度 θ，得到旋转后的向量 v_{rot} (见下图)

我们可以先通过点积计算 v 在 n 方向上的平行分量 $v_{||}$

$$v_{||} = (v \cdot n)n \tag{1}$$

再通过叉乘得到与n 正交的两个向量 $n_{\perp}$ 和 w

$$w = n \times v \tag{2}$$

$$n_{\perp} = v - v_{||} \tag{3}$$

从图中我们可以得到 v_{rot}

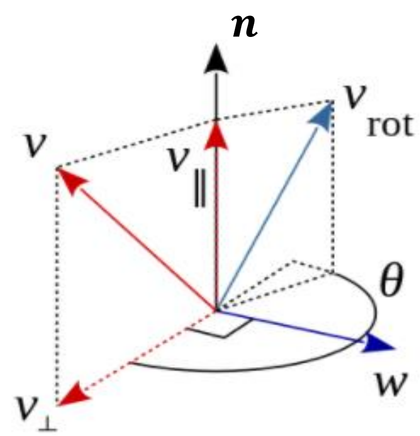
$$v_{rot} = v_{||} + \cos\theta \cdot n_{\perp} + \sin\theta \cdot w \tag{4}$$

将(1)(2)(3)带入(4), 得到

$$\begin{aligned} v_{rot} &= v_{||} + \cos\theta \cdot (v - v_{||}) + \sin\theta \cdot n \times v \\ &= \cos\theta \cdot v + (1 - \cos\theta) \cdot v_{||} + \sin\theta \cdot n \times v \\ &= \cos\theta \cdot v + (1 - \cos\theta) \cdot \underbrace{(v \cdot n)n}_{=n(v\cdot n)=n(n^T v)=nn^T v} + \sin\theta \cdot n \times v \\ &= (\cos\theta \cdot I + (1 - \cos\theta) nn^T + \sin\theta \cdot n\hat{})v \end{aligned}$$

所以将v 绕着方向 n 旋转角度 θ 到 v_{rot} 的旋转矩阵R为

$$R = \cos\theta \cdot I + (1 - \cos\theta) nn^T + \sin\theta \cdot n\hat{}$$



7. 四元数运算性质的验证

(第 5 题和第 7 题为计算题，故以手写作答)

8. *熟悉 C++11

1) 语句 std::sort(avec.begin(), avec.end(), [] (const A&a1, const A&a2) {return a1.index < a2.index;});
使用了 lambda 表达式

[](const A&a1, const A&a2) {return a1.index<a2.index;}

---这里 sort 函数用于遍历容器 avec，并对容器的成员变量使用匿名 lambda 函数, 每次调用 lambda 函数时都会传入容器 avec 内遍历到的成员变量的引用。由于判断条件为 a1.index<a2.index，故将 avec 内的成员变量以 A.index 的成员参数由小到大进行排列。

2) 语句 for (auto& a: avec) cout <<a.index << ” ”;
---这里使用了序列 for 循环，用于遍历容器 avec 内的成员变量的引用，对遍历到的成员变量输出其成员参数 index。
---还使用了自动类型推导 auto，可以从容器 avec 推断出容器的成员变量的类型为 A，相当于 for (A& a: avec) cout <<a.index << ” ”;

参考文献:

[1] Rodrigues' rotation formula
https://en.wikipedia.org/wiki/Rodrigues%27_rotation_formula
[2] 30 分钟了解 C++11 新特性