数据处理

RANSAC

RANSAC(Random Sample Consensus)是一种不确定算法。基本假设是数据是由**内点**和**外点**组成。**内点**就是组成产生模型的点，**外点**就是噪声或者离群点等非模型的点。

算法思想

1. 从数据集中随机选取模型所需的最少数量的数据，用这些数据建立模型
2. 将数据集中剩下的数据分别代入模型，若该数据满足模型（在误差范围内），则将这些数据加入模型中
3. 重复1,2两步骤，每次将数据点多、误差更小的模型留下供下次比较。
4. 迭代一定次数后产生最终结果（要么迭代次数到达上限，要么精度已经满足要求）

> k ---- 最大迭代次数
>
> n ---- 用于拟合的最小数据集大小
>
> t ---- 误差容许范围
>
> d ---- 模型中至少需要多少个数据点

```
/// 遍历
while iterations < k do
    maybeInliers := n /// 从数据中随机选择n个拟合的数据组
    maybeModel := model parameters fitted to maybeInliers /// 根据以上n个数据获得模型的参数
    alsoInliers := empty set  /// 初始化空数据组

    for every point in data not in maybeInliers do  /// 遍历：将处maybeInliers外的其他数据一一与模型进行比较
        if point fits maybeModel with an error smaller than t /// 如果比较下来误差在t范围内，则调价到inliers集合中
             add point to alsoInliers
    end for

    if the number of elements in alsoInliers is > d then /// 如果当前得到的Inliers集合中的数据量大于d
        // This implies that we may have found a good model ///意味着该模型是个“好”模型（即好参数）
        // now test how good it is.
        betterModel := model parameters fitted to all points in maybeInliers and alsoInliers
        thisErr := a measure of how well betterModel fits these points
        /// 如果当前模型的与Inliers中数据的误差比之前得到的最小误差更小，则更新最小误差，
        /// 最优模型参数设置为当前模型参数
        if thisErr < bestErr then
            bestFit := betterModel
            bestErr := thisErr
        end if
    end if
```

```
    increment iterations  /// 继续遍历
end while
```

卡尔曼滤波

b站视频链接

直观理解卡尔曼滤波（英语）

插值

已知$[a,b]$上n+1个不同点及其函数值(x_i,y_i)，$i=0,1,2,3...n$，求一个至多n次多项式：

$$\phi_n(x)=a_0+a_1x+...+a_nx^n$$

使得$\phi_n(x_i)=y_i$

拉格朗日插值

> https://www.zhihu.com/question/58333118/answer/262507694

插值多项式：

$$L_n(x)=\Sigma_{i=0}^n y_i l_i(x)=\Sigma_{i=0}^n y_i(\Pi_{j=0}^n\frac{(x-x_j)}{(x_i-x_j)})$$

牛顿插值

> https://www.zhihu.com/question/22320408/answer/141973314

差商

一阶差商 $f[x_i,x_j]=\frac{(f(x_i)-f(x_j))}{x_i-x_j}$

二阶差商$f[x_i,x_j,x_k]=\frac{f[x_i,x_j]-f[x_j,x_k]}{x_i-x_k}$

k阶差商$f[x_0,x_1,\dots,x_k]=\frac{f[x_0,x_1,\dots,x_{k-1}]-f[x_1,x_2,\dots,x_k]}{x_0-x_k}$

牛顿插值多项式

$$N_n(x)=f(x_0)+(x-x_0)f[x_0,x_1]+...+(x-x_0)(x-x_1)...(x-x_{n-1})f[x_0,x_1,\dots,x_n]$$

分段线性插值

每一段都认为是一次函数，将这些一次函数连起来即可

$$\phi_n(x)=\begin{cases}y_0,x<x_0\\ y_i+\frac{(x-x_i)(y_{i+1}-y_i)}{x_{i+1}-x_i},x_i\le x<x_{i+1}\\ y_n,x>x_n\end{cases}$$

最小二乘法

使观测值和理论值相差最小即可得到相应的函数。理论值是我们的多项式函数。相差大小的衡量是通过残差平方和来确定。

$$
\begin{gathered}
h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \ldots + \theta_n x^n, \\
\exists \boldsymbol{\theta} = (\theta_0, \theta_1 \ldots \theta_n) \\
s.t.\ \min(\Sigma_{i=1}^{n} (h_\theta(x_i) - y_i)^2)
\end{gathered}
$$

从**代数**角度，要使上述取最小值，即确定一组$\boldsymbol{\theta}$使上述表达式对这组值的偏导数等于零即可求出对应的值从而确定我们多项式函数的表达式。

从**矩阵**角度，上述$\min(\Sigma_{i=1}^{n} (h_\theta(x_i) - y_i)^2)$，从矩阵表达即$J(\theta) = (\boldsymbol{X}\theta - y)^T (\boldsymbol{X}\theta - y)$，为方便计算，通常将表达式变为$J(\theta) = \frac{1}{2} (\boldsymbol{X}\theta - y)^T (\boldsymbol{X}\theta - y)$，$h_\theta(x) = \boldsymbol{X}\theta$。矩阵求导后可得$\boldsymbol{\theta}$需要满足

$$
\theta = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T Y
$$

则得到多项式函数的各个参数的值