| | |
|---|---|
| **前沿计算研究实践（II）** | **2019 年春季** |

# 专题 2：最小二乘法

教授：陈宝权　　作者：袁野，袁昊琦

**免责声明**：该笔记尚未进行正式出版物的通常审查，未经教师允许不能向课程外分发。

## 2.1 最小二乘法的概念

**定义 2.1 最小二乘法** 将问题归约到优化一些平方项的和，再通过求偏导使我们得到一个线性系统，进而使用线性代数的方法求得线性系统的解，从而解决问题的方法。

我们为什么要学习最小二乘法呢?：（1）利于计算；（2）在图形的拟合中有很多应用。

## 2.2 最小二乘法应用实例

### 2.2.1 直线的拟合

#### 2.2.1.1 基本方法

- 给定：$n$ 个点 $\mathbf{p}_i = (x_i, y_i)$
- 目标：求出一条直线拟合这些点最好
- 问题的形式化：求出直线参数使得最小化到直线距离平方和

$$\arg\min_{a,b} E(a,b) := \sum_{i=1}^{n} (y_i - ax_i - b)^2 \tag{2.1}$$

- 求解方程 (2.1)：

$$\begin{aligned} \partial E(a,b)/\partial a &= \sum_{i=1}^{n} (-2x_i)(y_i - ax_i - b) = 0 \\ \partial E(a,b)/\partial b &= \sum_{i=1}^{n} (-2)(y_i - ax_i - b) = 0 \end{aligned} \tag{2.2}$$

化简式子 (2.2) 可以得到：

$$(\sum_{i=1}^{n} \begin{bmatrix} x_i^2 & x_i \\ x_i & 1 \end{bmatrix}) \begin{bmatrix} a \\ b \end{bmatrix} = \sum_{i=1}^{n} \begin{bmatrix} x_i y_i \\ y_i \end{bmatrix} \tag{2.3}$$

求解这个线性系统即可以解出方程

- 缺点：拟合直线会随坐标系变化而变化

#### 2.2.1.2 距离的另一种度量

- 详细内容会在下一章使用 PCA 方法着重介绍
- 最优化目标变为：

$$\arg\min_{\mathbf{n},d} E(\mathbf{n},d) = \sum_{i=1}^{n}(\mathbf{n}^T\mathbf{p}_i + d)^2, \quad s.t. \|n\| = 1 \tag{2.4}$$

首先对 $d$ 求偏导可以得到

$$\begin{aligned} &\partial E/\partial d = 0 \\ &\Rightarrow \sum_i(\mathbf{n}^T\mathbf{p}_i + d) = 0 \\ &\Rightarrow d = -\mathbf{n}^T\overline{\mathbf{p}} \end{aligned} \tag{2.5}$$

于是问题转化成最优化下式：

$$\arg\min_{\mathbf{n}} E(\mathbf{n}) = \sum_i(\mathbf{n}^T\mathbf{p}_i - \mathbf{n}^T\overline{\mathbf{p}}) = \sum_i(\mathbf{n}^T\widetilde{\mathbf{p}}_i), \quad s.t. \|n\| = 1 \tag{2.6}$$

我们利用拉格朗日乘数法，即求：

$$\arg\min_{\mathbf{n}}(\sum_i(\mathbf{n}^T\widetilde{\mathbf{p}}_i)^2 + \lambda(1 - \mathbf{n}^T\mathbf{n})) = \arg\min_{\mathbf{n}}(\mathbf{n}^TC\mathbf{n} + \lambda(1 - \mathbf{n}^T\mathbf{n})) \tag{2.7}$$

其中 $C = \sum_i \widetilde{\mathbf{p}}_i\widetilde{\mathbf{p}_i^T}$ 对 $\mathbf{n}$ 求导可以得到：

$$2C\mathbf{n} - 2\lambda\mathbf{n} = 0 \Rightarrow C\mathbf{n} = \lambda\mathbf{n} \tag{2.8}$$

将 (2.8) 结果带回 (2.7) 得到最小平方和为 $\lambda$，于是我们只需要找到对应最小特征值的特征向量 $\mathbf{n}$ 即可。

### 2.2.2 多项式拟合

- 问题描述：同样给定 $\mathbf{P} := \{\mathbf{p}_i\}$ 并先验地知道要拟合的函数为 $m$ 次：$f(x) = \sum_{k=0}^{n} a_k x^k$
- 问题被转化为：

$$\arg\min_{a_k} := \sum_{i=1}^{n}(f(x_i) - y_i)^2 = \sum_{i=1}^{n}(\sum_{k=0}^{m} a_k x_i^k - y_i)^2 \tag{2.9}$$

- 求解式子 (2.9)：

$$
\begin{aligned}
\partial E(a_0, a_1, ...)/\partial a_0 &= \sum_{i=1}^{n}(-2\partial f(x_i)/\partial a_0)(y_i - a_0 x_i - b) = 0 \\
\partial E(a_0, a_1, ...)/\partial a_1 &= \sum_{i=1}^{n}(-2\partial f(x_i)/\partial a_1)(y_i - a_1 x_i - b) = 0 \\
&... \\
\end{aligned}
\tag{2.10}
$$

化简得：

$$
(\sum_{i=1}^{n}\begin{bmatrix} x_i^{2m} & x_i^{2m-1} & ... & x_i^{m} \\ x_i^{2m-1} & x_i^{2m-2} & ... & x_i^{m-1} \\ ... & ... & ... & ... \\ x_i^{m} & x_i^{m-1} & ... & 1 \end{bmatrix})\begin{bmatrix} a_m \\ a_{m-1} \\ ... \\ a_0 \end{bmatrix} = \sum_{i=1}^{n}\begin{bmatrix} x_i^m y_i \\ x_i^{m-1} y_i \\ ... \\ y_i \end{bmatrix} \tag{2.11}
$$

再求解这个线性系统即可

- 利用同样的思路同样可以求解形如 $f_{\lambda_1,\lambda_2,...,\lambda_k}(x) = \sum_{i=1}^{k}\lambda_i f_i(x)$ 形式的拟合问题。

### 2.2.3 求解最小二乘意义下的线性系统

- 思考方程 $A\overrightarrow{x} = \overrightarrow{b}$，其中 $A$ 是一个 $m * n$ 的矩阵，如果 $m = n$ 并且 $A$ 满秩我们可以得到 $\overrightarrow{x} = A^{-1}\overrightarrow{b}$

- 当 $m > n$，那么有效方程数多于未知量个数，我们就可以去求最小二乘解（详细内容可以参考高等代数内容）。

- 如果 $A$ 的列向量是线性无关的，我们去求解最小二乘解，即求：

$$
\arg\min_{\overrightarrow{x}} \|A\overrightarrow{x} - \overrightarrow{b}\|^2 = \arg\min_{\overrightarrow{x}}(\overrightarrow{x}^T A^T A\overrightarrow{x} - 2\overrightarrow{x}^T A^T\overrightarrow{b} + \overrightarrow{b}^T\overrightarrow{b}) \tag{2.12}
$$

即求矩阵 $A$ 的列向量空间中与 $\overrightarrow{b}$ 距离最近的向量，于是使 $A\overrightarrow{a} - \overrightarrow{b}$ 与 $A$ 的列向量均垂直即可。即：

$$
\begin{aligned}
&\forall i, \overrightarrow{a_i}^T(A\overrightarrow{x} - \overrightarrow{b}) = 0 \\
&\Rightarrow A^T(A\overrightarrow{x} - \overrightarrow{b}) = 0 \\
&\Rightarrow A^T A\overrightarrow{x} = A^T\overrightarrow{b} \\
&\Rightarrow \overrightarrow{x} = (A^T A)^{-1}A^T\overrightarrow{b}
\end{aligned}
\tag{2.13}
$$

### 2.2.4 离群值问题

- 考虑在给定的点集中有一些离群点，它们因为某些原因不和其它点处于同一曲线上，那么将这些点计入拟合过程中会降低拟合效果。

- 解决方案：在优化直线参数时，给每一个点到直线的距离平方项加一个权重，即去优化下式：

$$\arg\min_{\mathbf{l}} E(\mathbf{l}) := \sum_{i=1}^{n} \omega_i d(\mathbf{l}, \mathbf{p}_i) \tag{2.14}$$

- 关于 $\omega_i$ 的选取，可以采取如下方案：进行多次迭代，并选取 $\omega_{j+1} := \exp(-d(\mathbf{l}_j, \mathbf{p}_i)\sigma)$，其中 $j$ 表示迭代次数，这样如果离群点数目比较少经过多次迭代离群点的权重就会逐渐变低，从而得到较好的拟合效果。

### 2.2.5 三维点的局部拟合

- 问题目标：给定三维空间的一个点集 $\mathbf{P} := \{\mathbf{p}_i\}$，并将其视为一个表面 $\mathcal{S}$ 的采样，下面我们希望可以在局部（即点 $\mathbf{p}$ 的 $r$ 范围邻域 $N_r(\mathbf{p}) = \{\mathbf{q}_i\}, \quad \mathbf{q}_i \in \mathbf{P}$）估计或者拟合这个表面。

- 方法：类似于上面的直线拟合的另一种距离估计，我们同样可以得到 $\mathbf{n}$ 应当是协方差矩阵 $\sum_i (\mathbf{q}_i - \overline{\mathbf{q}})(\mathbf{q}_i - \overline{\mathbf{q}})^T$ 的最小特征根对应的特征向量。

- 显然这种估计方法准确度取决于 $r$ 的选取，$r$ 过小信息量不足,$r$ 过大精细结构无法表示。

- 思考：如何选取 $r$ 的大小?

### 2.2.6 RANSAC

- 最小二乘法适用于有少量噪声点情况的最优化方法，但是如果噪声点非常多，我们希望能够找到我们的目标，无异于“大海捞针”。而 RANSAC(Random Sample Consensus) 便是处理这种情况的算法。

- 算法流程：
  （1）首先找一些点作为假设的局内点，通过这些点计算出模型的参数；
  （2）使用这个模型去测试其它数据，尝试找到跟这个模型匹配的其它局内点；
  （3）如果有足够多的局内点，那么我们就可以认为这个假设足够合理
  （4）用所有被归为局内点的点重新估计模型；
  （5）设计一个评估函数，使用（4）模型的错误率等去评估模型；
  （6）重复执行上述操作并限制总次数，每一次模型要么因为局内点太少而被舍弃，要么因为比现有模型好而被选用。

## 2.3 图形学中的应用

### 2.3.1 网格重建

**问题描述：** 输入三维空间中的一些边和控制点，重建物体表面的网格。
**形式化：** 网格中的顶点要满足平滑条件：$v_i - \frac{1}{d_i}\sum_{j:(i,j)\in E} v_j = 0$

**定义 2.2** 图的拉普拉斯算子 $L$:

$$L_{ij} = \begin{cases} 1 & i = j \\ -\frac{1}{d_i} & (i,j) \in E \\ 0 & \end{cases}$$

将网格中所有顶点的 $x, y, z$ 三个分量构成的向量记为 $\vec{x}, \vec{y}, \vec{z}$。拉普拉斯平滑条件即：

$$L\vec{x} = 0, L\vec{y} = 0, L\vec{z} = 0$$

记控制点为 $v_s = (x_s, y_s, z_s), s \in C = \{s_1, \cdots, s_m\}$，合并控制点条件和拉普拉斯平滑条件，有：

$$A\vec{x} = \vec{b}$$

其中

$$A = \begin{pmatrix} L \\ F \end{pmatrix}, F_{ij} = \begin{cases} 1 & j = s_i \in C \\ 0 & j \neq s_i \end{cases}$$

$$b_k = \begin{cases} 0 & k \leq n \\ x_{s_{k-n}} & n < k \leq n+m \end{cases}$$

**转化：**用最小二乘法求解

$$\min_{\vec{x}} ||A\vec{x} - \vec{b}||^2$$

### 2.3.2 表皮计算

通过顶点到相关骨架的加权，在动画、游戏中进行虚拟人物的表皮计算。

**问题描述：**构造一个顶点的变换 $T = (T_i)$ 用来估计输入的网格。S 个网格的序列顶点集构成向量 $P = (p^1, p^2, \cdots, p^S)$，其中 $p^t \in R^{3N}$。对每个 t 找一个变换 $T^t$ 用来估计 $p^t$：$p^t \approx T^t\tilde{p}$。$T^t$ 是一个线性组合：

$$T_i^t = \sum_{b\in B_i} w_{ib}\tilde{T}_b^t$$

**估计方法：** 对顶点 i 用最小二乘法选取 $\beta$ 个骨架：

$$\gamma_{bi} = \sum_{1 \le t \le S} ||p_i^t - \tilde{T}_b^t \tilde{p}_i||_2^2,\ b = 1, \cdots, \beta$$

用有约束的最小二乘估计顶点权值：

$$\sum_{b \in B_i} (\tilde{T}_b^t \tilde{p}_i) w_{ib} = p_i^t,\ t = 1, \cdots, S$$

$$s.t.\ \sum_b w_{ib} = 1$$

记为

$$A^{(i)} w^{(i)} = b^{(i)}$$

**减轻过拟合：** 用优化的线性系统避免过拟合

$$\begin{pmatrix} cA^{(i)} \\ 1 \cdots 1 \end{pmatrix} w^{(i)} = \begin{pmatrix} cb^{(i)} \\ 1 \end{pmatrix}$$

其中 c 是比例因子

$$c = \frac{1}{||b^{(i)}||_2}$$

# 参考文献

[Daniel Cohen-Or] , A Sampler of Useful Computational Tools for Applied Geometry, Computer Graphics, and Image Processing (1997), pp. 1–12.

[张贤科，许甫华] ，《高等代数学》pp.116-117