局部线性回归推导

1. 问题定义

给定训练数据集 $\{(x_i, y_i)\}_{i=1}^m$ ，其中 $x_i \in \mathbb{R}^n$ ， $y_i \in \mathbb{R}$。对于查询点 x ，预测其输出值 $\hat{y}$ 。

2. 目标函数

局部加权线性回归的目标函数为加权残差平方和：

$$J(\theta) = \sum_{i=1}^{m} w^{(i)} \left(y^{(i)} - \theta^T x^{(i)} \right)^2$$

其中：

- $\theta \in \mathbb{R}^{n+1}$ 是待求解的参数向量（包含截距项）
- $w^{(i)}$ 是样本点 $x^{(i)}$ 的权重，由核函数计算： $w^{(i)} = \exp\left(-\frac{\|x^{(i)} - x\|^2}{2\tau^2}\right)$
- $\tau > 0$ 是带宽参数（bandwidth），控制权重衰减速度

3. 矩阵形式推导

3.1 定义矩阵

- **设计矩阵** $X \in \mathbb{R}^{m \times (n+1)}$ ：

$$X = \begin{bmatrix} 1 & x_1^{(1)} & \cdots & x_n^{(1)} \\ 1 & x_1^{(2)} & \cdots & x_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(m)} & \cdots & x_n^{(m)} \end{bmatrix}$$

- **权重矩阵** $W \in \mathbb{R}^{m \times m}$ ：

$$W = \operatorname{diag}\left(w^{(1)}, w^{(2)}, \ldots, w^{(m)}\right) = \begin{bmatrix} w^{(1)} & 0 & \cdots & 0 \\ 0 & w^{(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w^{(m)} \end{bmatrix}$$

- **目标向量** $y \in \mathbb{R}^m$ ：

$$y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

3.2 目标函数的矩阵形式

$$J(\theta) = (y - X\theta)^T W (y - X\theta)$$

3.3 求解最优参数 θ

对 $J(\theta)$ 求梯度并令其为零：

$$\nabla_\theta J(\theta) = -2X^T W(y - X\theta) = 0$$

整理得正规方程：

$$X^T W X\theta = X^T W y$$

解得最优参数（熟悉不？正规方程权重版）：

$$\boxed{\theta = (X^T W X)^{-1} X^T W y}$$

4. 预测查询点输出

对于查询点 x （需增广为 $[1, x_1, \ldots, x_n]^T$ ），预测值为：

$$\boxed{\hat{y} = \theta^T x = x^T (X^T W X)^{-1} X^T W y}$$

5. 权重函数详解

5.1 高斯核函数

$$w^{(i)} = \exp\left(-\frac{\|x^{(i)} - x\|^2}{2\tau^2}\right)$$

- $\|x^{(i)} - x\|^2 = \sum_{k=1}^{n} (x_k^{(i)} - x_k)^2$
- τ 的作用：
  - $\tau \uparrow$：权重衰减慢，模型更平滑（欠拟合倾向）
  - $\tau \downarrow$：权重衰减快，模型更复杂（过拟合倾向）

高斯核怎么得到的？看看AI：高斯核推导，如果你了解最小二乘法的由来，那么高斯核也好理解

5.2 其他常用核函数（仅作了解）

- **Epanechnikov 核**：

$$w^{(i)} = \max\left(1 - \frac{\|x^{(i)} - x\|^2}{\tau^2}, 0\right)$$

- **Tricube 核：**

$$w^{(i)} = \left(1 - \left|\frac{x^{(i)} - x}{\tau}\right|^3\right)^3 \cdot \mathbf{1}_{\left|\frac{x^{(i)} - x}{\tau}\right| < 1}$$

6. 算法复杂度分析

- **时间复杂度：** $O(mn^2 + n^3)$
  - 每预测一个新点需重新计算权重矩阵 W 和矩阵逆 $(X^T W X)^{-1}$
  - 当 $m \gg n$ 时，复杂度主要由 $O(mn^2)$ 主导
- **空间复杂度：** $O(mn + m)$
  - 需存储设计矩阵 X 和权重矩阵 W

7. 与标准线性回归对比

特性	**标准线性回归**	**局部加权线性回归**
模型形式	全局线性模型	局部线性模型
参数	θ	θ （每点重新计算）
权重	均匀权重	核函数加权
拟合能力	只能拟合线性关系	可拟合非线性关系
计算效率	$O(n^3)$ （一次性求解）	$O(mn^2)$ （每点计算）
参数更新	无需重新训练	每预测一点需重新训练

8. 正则化改进

为防止 $X^T W X$ 不可逆，加入正则化项（类似岭回归）：

$$\theta = (X^T W X + \lambda I)^{-1} X^T W y$$

其中 $\lambda > 0$ 是正则化系数， I 为单位矩阵。

9. 伪代码

代码块

```
def LOWESS_predict(X_train, y_train, x_query, tau):
    m = len(X_train)   # 样本数
    n = len(x_query)   # 特征维数
    W = np.zeros((m, m))

```

```python
    # 计算权重矩阵
    for i in range(m):
        dist = np.linalg.norm(X_train[i] - x_query)
        W[i, i] = np.exp(-dist**2 / (2 * tau**2))

    # 增广设计矩阵
    X_aug = np.hstack([np.ones((m, 1)), X_train])

    # 求解正规方程
    A = X_aug.T @ W @ X_aug + 1e-6 * np.eye(n+1)  # 加入正则项防奇异
    b = X_aug.T @ W @ y_train
    theta = np.linalg.solve(A, b)

    # 预测查询点
    x_aug_query = np.hstack([1, x_query])
    return x_aug_query @ theta
```

10. 关键公式总结

1. **权重计算**：

$$w^{(i)} = \exp\left(-\frac{\|x^{(i)} - x\|^2}{2\tau^2}\right)$$

2. **参数估计**：

$$\theta = (X^T W X)^{-1} X^T W y$$

3. **预测输出**：

$$\hat{y} = x^T (X^T W X)^{-1} X^T W y$$

此推导完整展示了局部加权线性回归的理论基础、计算流程和实现细节。