# 1 The Derivative Explanation of Formulas 2.9 to 2.13

之前我们得到了在线性回归中关于 $f(x) = E[Y|X]$ 的推导。

下面我们再看 KNN 算法，KNN 算法尝试通过训练数据直接通过训练数据完成任务，对于每一个给定点 $x$, 我们需要对所有的输入变量 $x_i = x$ 求其对应 $y_i$ 的均值

$$\hat{f}(x) = Ave(y_i|x_i \in Neighbor(x)) \tag{1}$$

在上式中，$Ave$ 是求平均值的函数，$Neighbor(x)$ 是一个包含 k 个与 x 距离最近的点的领域 ($Neighbor(x)$ is a Neighbor containing the k points in T closest to x)

这里使用了两个近似

- 期望被近似为对所有样本点求均值
- 一个给定点的条件期望被”松弛”为离该点距离近的区域上的条件期望

当样本数据量很大的时候，$x$ 周围的点会非常靠近 $x$. 同时，根据 mild regularity conditions, 可以证明当 $N, k -> \infty$ 同时 $\frac{k}{N} -> 0$, 我们可以得到 $\hat{f}(x) -> E[Y|X = x]$

同样对于线性回归模型

$$L(y, X\beta) = (y - X\beta)^2 \tag{2}$$

$$EPE(\beta) = \int_{x,y} (y - X\beta)^2 Pr(X = x, Y = y) dxdy \tag{3}$$

$$\begin{aligned} \frac{\partial EPE(\beta)}{\partial \beta} &= \int_x y - X^T(y - X\beta) Pr(X = x, Y = y) dxdy \\ &= \beta E[X^T X] - E[X^T y] \end{aligned} \tag{4}$$

we set equation (4) to 0, we get $\beta = [E[X^T X]]^{-1} E[X^T y]$ 我们将这个表达式与之前的推导式 $\hat{\beta} = (X^T X)^{-1} X^T y$ 进行对比, 最小二乘法的解相当于用训练数据的平均值替换掉了 (4) 中的期望

所以，k-nearest neighbor 和最小二乘法都是通过求平均来近似代替条件期望，但是他们在模型的假设上具有很大的不同

- 最小二乘法假定 $f(x)$ 能被一个全局的线性函数较好的拟合
- k-nearest neighbor 假定 $f(x)$ 能被一个 locally constant function 较好的近似

如果我们使用 $L_1$ 损失函数代替 $L_2$, 在这种情况下，解答是条件中位数 (conditional median)

$$\hat{f}(x) = median(Y|X = x) \tag{5}$$

如果输出变量是一个分类变量呢？我们依然可以使用同一个范式进行处理，除了使用一个不同的损失函数。一个估计变量 $\hat{G}$，损失函数可以被表示为一个 $K \times K$ 的矩阵 $L$，其中 $L$ 的对角元素为 0，非对角元素为非负整数，其中 $L(k,l)$ 是将原本是 $G_l$ 的类错分为 $G_k$ 类所付出的代价。同样我们可以先求出 expected prediction error:

$$EPE = E[L(G, \hat{G}(X))] \tag{6}$$

同样，上式的期望依赖于联合概率分布 $Pr(G, X)$

$$\begin{aligned} EPE &= \int_{i=1}^{k} \int_{j=1}^{n} L(G_i, \hat{G}(x_j)) Pr(G_k = m, X = x_j) && (7) \\ &= E_X \int_{k=1}^{K} L(G_k, \hat{G}(X)) Pr(G_k|X) && (8) \end{aligned}$$

可以注意到，期望里面的部分始终非负，为了最小化整个期望，我们可以最小化期望里面的部分

$$\begin{aligned} \hat{G}(x) &= \arg\min_{g \in G} \int_{k=1}^{K} L(G_k, g) Pr(G_k|X = x) && (9) \\ &= \arg\min_{g \in G} [1 - Pr(g|X = x)] && (10) \end{aligned}$$

(10) 式可以改写为 $\hat{G}(X) = G_k \ if \ Pr(G_k|X = x) = \max_{g \in G} Pr(g|X = x)$
这个合理的结果为成为 Bayes classifier.

# 2 高维问题的局部方法

我们已经证明了两种在预测预测问题中的学习策略：稳定但是带有偏差的线性模型，不太稳定但是具有较小偏差的 k-最近邻估计。

那么我们可能会认为, 只要给予足够大的训练数据，k-nearest 算法理论上就能够逼近最优解 (条件期望)，但是这种方法在高维情况下就会失败

下面我们将要讨论维数灾难 (curse of dimensionality)。主要有两点

- 如果我们想使用 k-nearest 算法来获得 neighbor 中的点，其与整个空间中的点的比例为 r，由于空间与线段的关系，想要获得一定比例的数据，我们需要更长的线段来满足这一条件，比如在一个边长为 1 的十维的超立方体中，我们想要获得 1% 的数据，需要付出 0.63 的边长。
- 另外，如果在一个高维空间中进行稀疏取样，会使得所有的样本点离样本的某一边很近

# 3 统计模型，监督学习和函数逼近

我们的目标在于找到一个对 $f(x)$ 的有效逼近 $\hat{f}(x)$，能够有效的表示出输入和输出之间的映射关系。从之前的章节可以知道，对于变量连续的情况下，回归函数 $f(x) = E(Y|X = x)$。最近邻算法可以被看作是对这种条件期望的一种直接估计。但是我们看到了，至少在两种情况下，该方法会失败

- 如果输入空间的维数太高，会使得最近邻不需要离目标点距离近，导致较大的误差。
- If special structure is known to exist, this can be used to reduce both the bias and the variance of the estimates. (不懂)