神经网络预备知识

金曙松

溢思得瑞集团

2017 年 3 月 10 日

机器学习的主要目的 — 划线

- 机器学习的主要问题
  1. 分类 (classification)
  2. 预测 (prediction)
- 以上两大任务都归结为划线 (直线，曲线，环线等等)

机器学习的主要目的 — 划线

- 机器学习的主要问题
  1. 分类 (classification)
  2. 预测 (prediction)
- 以上两大任务都归结为划线 (直线，曲线，环线等等)

机器学习的主要目的 — 划线

- 机器学习的主要问题
  1. 分类 (classification)
  2. 预测 (prediction)
- 以上两大任务都归结为划线 (直线，曲线，环线等等)

机器学习的主要目的 — 划线

- 机器学习的主要问题
  1. 分类 (classification)
  2. 预测 (prediction)
- 以上两大任务都归结为划线 (直线，曲线，环线等等)

两大类别：有监督和无监督学习

- 有监督学习：数据有标签；回归
- 无监督学习：数据无标签；聚类

两大类别：有监督和无监督学习

- 有监督学习：数据有标签；回归
- 无监督学习：数据无标签；聚类

有监督学习

- Y：输出结果，也加响应变量，目标等等
- X: 预测变量，一般是个 p 维向量，也叫输入 (inputs)、协变量 (covariates)、属性 (features)、自变量 (independent variables)
- 在回归问题中，Y 是定量的 (如价格，血压等)
- 在分类问题中，Y 是定性的，而且一般都只有有限类。这些类一般无序 (如生存/死亡、0-9 到底手写的是什么数字、这些症状到底是什么疾病等等)
- 我们有的是训练数据 $(x_1, y_1) \ldots, (x_N, y_N)$。他们又称为样本，实例等。

有监督学习

- Y：输出结果，也加响应变量，目标等等
- X: 预测变量，一般是个 p 维向量，也叫输入 (inputs)、协变量 (covariates)、属性 (features)、自变量 (independent variables)
- 在回归问题中，Y 是定量的 (如价格，血压等)
- 在分类问题中，Y 是定性的，而且一般都只有有限类。这些类一般无序 (如生存/死亡、0-9 到底手写的是什么数字、这些症状到底是什么疾病等等)
- 我们有的是训练数据 $(x_1, y_1) \ldots, (x_N, y_N)$。他们又称为样本，实例等。

有监督学习

- Y：输出结果，也加响应变量，目标等等
- X: 预测变量，一般是个 p 维向量，也叫输入 (inputs)、协变量 (covariates)、属性 (features)、自变量 (independent variables)
- 在回归问题中，Y 是定量的 (如价格，血压等)
- 在分类问题中，Y 是定性的，而且一般都只有有限类。这些类一般无序 (如生存/死亡、0-9 到底手写的是什么数字、这些症状到底是什么疾病等等)
- 我们有的是训练数据 $(x_1, y_1) \ldots, (x_N, y_N)$。他们又称为样本，实例等。

有监督学习

- Y：输出结果，也加响应变量，目标等等
- X: 预测变量，一般是个 p 维向量，也叫输入 (inputs)、协变量 (covariates)、属性 (features)、自变量 (independent variables)
- 在回归问题中，Y 是定量的 (如价格，血压等)
- 在分类问题中，Y 是定性的，而且一般都只有有限类。这些类一般无序 (如生存/死亡、0-9 到底手写的是什么数字、这些症状到底是什么疾病等等)
- 我们有的是训练数据 $(x_1, y_1) \ldots, (x_N, y_N)$。他们又称为样本，实例等。

- Y：输出结果，也加响应变量，目标等等
- X: 预测变量，一般是个 p 维向量，也叫输入 (inputs)、协变量 (covariates)、属性 (features)、自变量 (independent variables)
- 在回归问题中，Y 是定量的 (如价格，血压等)
- 在分类问题中，Y 是定性的，而且一般都只有有限类。这些类一般无序 (如生存/死亡、0-9 到底手写的是什么数字、这些症状到底是什么疾病等等)
- 我们有的是训练数据 $(x_1, y_1) \ldots, (x_N, y_N)$。他们又称为样本，实例等。

目的

- 精确地对未知情形给予预报
- 了解输入变量到底如何影响输出变量
- 对我们预报结果的质量进行合理的评估及给予推断 (例如预报的方差，置信区间等)

目的

- 精确地对未知情形给予预报
- 了解输入变量到底如何影响输出变量
- 对我们预报结果的质量进行合理的评估及给予推断 (例如预报的方差，置信区间等)

目的

- 精确地对未知情形给予预报
- 了解输入变量到底如何影响输出变量
- 对我们预报结果的质量进行合理的评估及给予推断 (例如预报的方差，置信区间等)

无监督学习

- 没有输出变量，仅有各种属性 (features)
- 目的更模糊 — 例如找到表现相似的族群等
- 对得到的结果也比较难评估
- 有时可以作为有监督学习的前期准备

无监督学习

- 没有输出变量，仅有各种属性 (features)
- 目的更模糊 — 例如找到表现相似的族群等
- 对得到的结果也比较难评估
- 有时可以作为有监督学习的前期准备

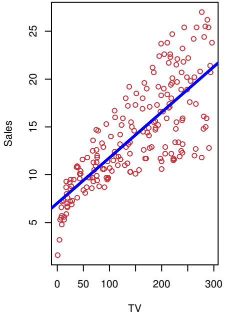
无监督学习

- 没有输出变量，仅有各种属性 (features)
- 目的更模糊 — 例如找到表现相似的族群等
- 对得到的结果也比较难评估
- 有时可以作为有监督学习的前期准备

无监督学习

- 没有输出变量，仅有各种属性 (features)
- 目的更模糊 — 例如找到表现相似的族群等
- 对得到的结果也比较难评估
- 有时可以作为有监督学习的前期准备

一个例子



图中显示的是销售额 (Sales) 与电视 (TV)、电台 (Radio) 和报纸 (Newspaper) 广告的关系。蓝色的线是 Sales 对 TV、Radio 和 Newspaper 的线性回归。我们如果用三个属性 (features)，是否结果会好一些？于是，模型为

$$\mathrm{Sales} \approx \mathrm{f}(\mathrm{TV}, \mathrm{Radio}, \mathrm{Newspaper})$$

记号

- Sales 在这里是目标、也叫响应变量，用 Y 表示
- TV 记为 X_1，Radio 和 Newspaper 记为 X_2, X_3
- 于是用向量表示为

$$X = \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix}$$

- 于是模型为

$$Y = f(X) + \varepsilon$$

其中 ε 用于表示无法被模型描述的误差和随即扰动。

记号

- Sales 在这里是目标、也叫响应变量，用 Y 表示
- TV 记为 X_1，Radio 和 Newspaper 记为 X_2, X_3
- 于是用向量表示为

$$X = \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix}$$

- 于是模型为

$$Y = f(X) + \varepsilon$$

其中 ε 用于表示无法被模型描述的误差和随即扰动。

记号

- Sales 在这里是目标、也叫响应变量，用 Y 表示
- TV 记为 X_1，Radio 和 Newspaper 记为 X_2, X_3
- 于是用向量表示为

$$X = \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix}$$

- 于是模型为

$$Y = f(X) + \varepsilon$$

其中 ε 用于表示无法被模型描述的误差和随即扰动。

记号

- Sales 在这里是目标、也叫响应变量，用 Y 表示
- TV 记为 X_1，Radio 和 Newspaper 记为 X_2, X_3
- 于是用向量表示为

$$X = \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix}$$

- 于是模型为

$$Y = f(X) + \varepsilon$$

其中 ε 用于表示无法被模型描述的误差和随即扰动。

回归函数 $f(\cdot)$

- $f(\cdot)$ 有各种选择，如何评价
- 在均方误差最小的原则下，我们选择

$$f(x) = E(Y|X = x)$$

- 这里 $f(x)$ 称为回归函数。

回归函数 $f(\cdot)$

- $f(\cdot)$ 有各种选择，如何评价
- 在均方误差最小的原则下，我们选择

$$f(x) = E(Y|X = x)$$

- 这里 $f(x)$ 称为回归函数。

回归函数 $f(\cdot)$

- $f(\cdot)$ 有各种选择，如何评价
- 在均方误差最小的原则下，我们选择

$$f(x) = E(Y|X = x)$$

- 这里 $f(x)$ 称为回归函数。

回归函数 $f(\cdot)$

根据 $f(\cdot)$ 的不同形式，有

- **线性回归** f 是个线性函数
- **非线性回归** f 是非线性函数，如 logistic 回归
- **非参数回归** f 是非参数函数，如局部多项式回归、核回归、样条等

本质上、传统的神经网络是非线性回归模型及其组合。

回归函数 f(·)

根据 f(·) 的不同形式，有

线性回归 f 是个线性函数

非线性回归 f 是非线性函数，如 logistic 回归

非参数回归 f 是非参数函数，如局部多项式回归、核回归、样条等

本质上、传统的神经网络是非线性回归模型及其组合。

回归函数 f(·)

根据 f(·) 的不同形式，有

- 线性回归 f 是个线性函数
- 非线性回归 f 是非线性函数，如 logistic 回归
- 非参数回归 f 是非参数函数，如局部多项式回归、核回归、样条等

本质上、传统的神经网络是非线性回归模型及其组合。

回归函数 f(·)

- 在上例中

$$f(x) = f(x_1, x_2, x_3) = E(Y|X_1 = x_1, X_2 = x_2, X_3 = x_3)$$

- 在均方误差的准则下，我们希望 f

$$E((Y - f(X))^2|X = x) \leqslant E((Y - g(X))^2|X = x) \tag{1}$$

  但这是做不到的。
- 必须限制 g 的类，使得 $f \in \{g\}$，且(1)成立。
- 记这样的 f 为 $\hat{f}$，于是

$$E[(Y - \hat{f}(X))^2|X = x] = [f(x) - \hat{f}(x)]^2 + var(\varepsilon)$$

  第一部分称为偏差的平方，第二部分为方差。

回归函数 f(·)

- 在上例中

$$f(x) = f(x_1, x_2, x_3) = E(Y|X_1 = x_1, X_2 = x_2, X_3 = x_3)$$

- 在均方误差的准则下，我们希望 f

$$E((Y - f(X))^2|X = x) \leqslant E((Y - g(X))^2|X = x) \tag{1}$$

但这是做不到的。

- 必须限制 g 的类，使得 $f \in \{g\}$，且(1)成立。
- 记这样的 f 为 $\hat{f}$，于是

$$E[(Y - \hat{f}(X))^2|X = x] = [f(x) - \hat{f}(x)]^2 + var(\varepsilon)$$

第一部分称为偏差的平方，第二部分为方差。

回归函数 $f(\cdot)$

- 在上例中

$$f(x) = f(x_1, x_2, x_3) = E(Y|X_1 = x_1, X_2 = x_2, X_3 = x_3)$$

- 在均方误差的准则下，我们希望 f

$$E((Y - f(X))^2|X = x) \leqslant E((Y - g(X))^2|X = x) \tag{1}$$

但这是做不到的。

- 必须限制 g 的类，使得 $f \in \{g\}$，且(1)成立。
- 记这样的 f 为 $\hat{f}$，于是

$$E[(Y - \hat{f}(X))^2|X = x] = [f(x) - \hat{f}(x)]^2 + \mathrm{var}(\varepsilon)$$

第一部分称为偏差的平方，第二部分为方差。

回归函数 f(·)

- 在上例中

$$f(x) = f(x_1, x_2, x_3) = E(Y|X_1 = x_1, X_2 = x_2, X_3 = x_3)$$

- 在均方误差的准则下，我们希望 f

$$E((Y - f(X))^2|X = x) \leqslant E((Y - g(X))^2|X = x) \tag{1}$$

  但这是做不到的。
- 必须限制 g 的类，使得 $f \in \{g\}$，且(1)成立。
- 记这样的 f 为 $\hat{f}$，于是

$$E[(Y - \hat{f}(X))^2|X = x] = [f(x) - \hat{f}(x)]^2 + var(\varepsilon)$$

  第一部分称为偏差的平方，第二部分为方差。

投影寻踪回归 (Projection Pursuit Regression)

- 要了解神经网络，最好先了解 logisitic 回归和 PPR
- 令 $\omega_m, m = 1, 2, \ldots, M$ 是一个 p 维的未知参数向量。为保证模型的可识别性，我们要求 ω_m 的模为 1;
- PPR 有如下形式：

$$f(X) = \sum_{m=1}^{M} g_m(\omega_m^T X) \tag{2}$$

投影寻踪回归 (Projection Pursuit Regression)

- 要了解神经网络，最好先了解 logisitic 回归和 PPR
- 令 $\omega_m, m = 1, 2, \ldots, M$ 是一个 p 维的未知参数向量。为保证模型的可识别性，我们要求 ω_m 的模为 1;
- PPR 有如下形式：

$$f(X) = \sum_{m=1}^{M} g_m(\omega_m^T X) \tag{2}$$

投影寻踪回归 (Projection Pursuit Regression)

- 要了解神经网络，最好先了解 logisitic 回归和 PPR
- 令 $\omega_m, m = 1, 2, \ldots, M$ 是一个 p 维的未知参数向量。为保证模型的可识别性，我们要求 ω_m 的模为 1;
- PPR 有如下形式：

$$f(X) = \sum_{m=1}^{M} g_m(\omega_m^T X) \tag{2}$$

说明

- 上述模型是一个加法模型 (additive model)
- 模型的自变量不是单变量，而是一个导出量：$V_m = \omega_m^T X$
- 函数 g_m 是未知的函数，需要按照 ω_m 的方向用非参数方法进行估计
- $g_m(\omega_m^T X)$ 称为 $\mathbb{R}^p$ 上的岭函数 (ridge function)，它只在由 ω_m 定义的方向上变化
- V_m 是 X 在 ω_m 方向上的投影
- 我们通过搜索 ω_m 使得模型与数据拟合得足够好
- M 如果足够大，通过选择 g_m，可以将任意在 $\mathbb{R}^p$ 空间中的连续函数拟合得任意精确
- 如果 $M = 1$，该模型称为单指标模型 (single index model)

说明

- 上述模型是一个加法模型 (additive model)
- 模型的自变量不是单变量，而是一个导出量：$V_m = \omega_m^T X$
- 函数 g_m 是未知的函数，需要按照 ω_m 的方向用非参数方法进行估计
- $g_m(\omega_m^T X)$ 称为 $\mathbb{R}^p$ 上的岭函数 (ridge function)，它只在由 ω_m 定义的方向上变化
- V_m 是 X 在 ω_m 方向上的投影
- 我们通过搜索 ω_m 使得模型与数据拟合得足够好
- M 如果足够大，通过选择 g_m，可以将任意在 $\mathbb{R}^p$ 空间中的连续函数拟合得任意精确
- 如果 $M = 1$，该模型称为单指标模型 (single index model)

说明

- 上述模型是一个加法模型 (additive model)
- 模型的自变量不是单变量，而是一个导出量：$V_m = \omega_m^T X$
- 函数 g_m 是未知的函数，需要按照 ω_m 的方向用非参数方法进行估计
- $g_m(\omega_m^T X)$ 称为 $\mathbb{R}^p$ 上的岭函数 (ridge function)，它只在由 ω_m 定义的方向上变化
- V_m 是 X 在 ω_m 方向上的投影
- 我们通过搜索 ω_m 使得模型与数据拟合得足够好
- M 如果足够大，通过选择 g_m，可以将任意在 $\mathbb{R}^p$ 空间中的连续函数拟合得任意精确
- 如果 $M = 1$，该模型称为单指标模型 (single index model)

说明

- 上述模型是一个加法模型 (additive model)
- 模型的自变量不是单变量，而是一个导出量：$V_m = \omega_m^T X$
- 函数 g_m 是未知的函数，需要按照 ω_m 的方向用非参数方法进行估计
- $g_m(\omega_m^T X)$ 称为 $\mathbb{R}^p$ 上的岭函数 (ridge function)，它只在由 ω_m 定义的方向上变化
- V_m 是 X 在 ω_m 方向上的投影
- 我们通过搜索 ω_m 使得模型与数据拟合得足够好
- M 如果足够大，通过选择 g_m，可以将任意在 $\mathbb{R}^p$ 空间中的连续函数拟合得任意精确
- 如果 $M = 1$，该模型称为单指标模型 (single index model)

说明

- 上述模型是一个加法模型 (additive model)
- 模型的自变量不是单变量，而是一个导出量：$V_m = \omega_m^T X$
- 函数 g_m 是未知的函数，需要按照 ω_m 的方向用非参数方法进行估计
- $g_m(\omega_m^T X)$ 称为 $\mathbb{R}^p$ 上的岭函数 (ridge function)，它只在由 ω_m 定义的方向上变化
- V_m 是 X 在 ω_m 方向上的投影
- 我们通过搜索 ω_m 使得模型与数据拟合得足够好
- M 如果足够大，通过选择 g_m，可以将任意在 $\mathbb{R}^p$ 空间中的连续函数拟合得任意精确
- 如果 $M = 1$，该模型称为单指标模型 (single index model)

说明

- 上述模型是一个加法模型 (additive model)
- 模型的自变量不是单变量，而是一个导出量：$V_m = \omega_m^T X$
- 函数 g_m 是未知的函数，需要按照 ω_m 的方向用非参数方法进行估计
- $g_m(\omega_m^T X)$ 称为 $\mathbb{R}^p$ 上的岭函数 (ridge function)，它只在由 ω_m 定义的方向上变化
- V_m 是 X 在 ω_m 方向上的投影
- 我们通过搜索 ω_m 使得模型与数据拟合得足够好
- M 如果足够大，通过选择 g_m，可以将任意在 $\mathbb{R}^p$ 空间中的连续函数拟合得任意精确
- 如果 $M = 1$，该模型称为单指标模型 (single index model)

说明

- 上述模型是一个加法模型 (additive model)
- 模型的自变量不是单变量，而是一个导出量：$V_m = \omega_m^T X$
- 函数 g_m 是未知的函数，需要按照 ω_m 的方向用非参数方法进行估计
- $g_m(\omega_m^T X)$ 称为 $\mathbb{R}^p$ 上的岭函数 (ridge function)，它只在由 ω_m 定义的方向上变化
- V_m 是 X 在 ω_m 方向上的投影
- 我们通过搜索 ω_m 使得模型与数据拟合得足够好
- M 如果足够大，通过选择 g_m，可以将任意在 $\mathbb{R}^p$ 空间中的连续函数拟合得任意精确
- 如果 $M = 1$，该模型称为单指标模型 (single index model)

说明

- 上述模型是一个加法模型 (additive model)
- 模型的自变量不是单变量，而是一个导出量：$V_m = \omega_m^T X$
- 函数 g_m 是未知的函数，需要按照 ω_m 的方向用非参数方法进行估计
- $g_m(\omega_m^T X)$ 称为 $\mathbb{R}^p$ 上的岭函数 (ridge function)，它只在由 ω_m 定义的方向上变化
- V_m 是 X 在 ω_m 方向上的投影
- 我们通过搜索 ω_m 使得模型与数据拟合得足够好
- M 如果足够大，通过选择 g_m，可以将任意在 $\mathbb{R}^p$ 空间中的连续函数拟合得任意精确
- 如果 $M = 1$，该模型称为单指标模型 (single index model)

如何估计 PPR

- 训练样本 (x_i, y_i)，$i = 1, \ldots, N$
- 损失函数

$$\sum_{i=1}^{N}\left[y_i - \sum_{m=1}^{M} g_m(\omega_m^T x_i)\right]^2 \tag{3}$$

- 以下假设 $M = 1$，舍弃脚标 m 以简化公式。其实，对 $M > 1$ 算法类似。

如何估计 PPR

- 训练样本 (x_i, y_i)，$i = 1, \ldots, N$
- 损失函数

$$\sum_{i=1}^{N}\left[y_i - \sum_{m=1}^{M} g_m(\omega_m^T x_i)\right]^2 \tag{3}$$

- 以下假设 $M = 1$，舍弃脚标 m 以简化公式。其实，对 $M > 1$ 算法类似。

如何估计 PPR

- 训练样本 (x_i, y_i)，$i = 1, \ldots, N$
- 损失函数

$$\sum_{i=1}^{N} \left[y_i - \sum_{m=1}^{M} g_m(\omega_m^T x_i) \right]^2 \tag{3}$$

- 以下假设 $M = 1$，舍弃脚标 m 以简化公式。其实，对 $M > 1$ 算法类似。

估计 g

- $V_i = \omega^T x_i$，注意 $M = 1$，因此，此处的 i 只表示样本编号，$i = 1, \ldots, N$;
- 可以用任意的非参数方法估计 g (例如局部多项式，核估计，样条等)
- 另一方面，当 g 给定后，通过搜索 ω，最小化(3)
- 这一般通过 Gauss-Newton 法进行搜索

估计 g

- $V_i = \omega^T x_i$，注意 $M = 1$，因此，此处的 i 只表示样本编号，$i = 1, \ldots, N$；
- 可以用任意的非参数方法估计 g (例如局部多项式，核估计，样条等)
- 另一方面，当 g 给定后，通过搜索 ω，最小化(3)
- 这一般通过 Gauss-Newton 法进行搜索

估计 g

- $V_i = \omega^T x_i$，注意 $M = 1$，因此，此处的 i 只表示样本编号，$i = 1, \ldots, N$;
- 可以用任意的非参数方法估计 g (例如局部多项式，核估计，样条等)
- 另一方面，当 g 给定后，通过搜索 ω，最小化(3)
- 这一般通过 Gauss-Newton 法进行搜索

估计 g

- $V_i = \omega^T x_i$，注意 $M = 1$，因此，此处的 i 只表示样本编号，$i = 1, \ldots, N$;
- 可以用任意的非参数方法估计 g (例如局部多项式，核估计，样条等)
- 另一方面，当 g 给定后，通过搜索 ω，最小化(3)
- 这一般通过 Gauss-Newton 法进行搜索

算法

- 令 ω_{old} 为现在对 ω 的估计，

$$g(\omega^T x_i) \approx g(\omega_{\mathrm{old}}^T x_i) + g'(\omega_{\mathrm{old}}^T x_i)(\omega - \omega_{\mathrm{old}})^T x_i \tag{4}$$

- 代入(3)，得到

$$\sum_{i=1}^{N}\left[y_i - g(\omega^T x_i)\right]^2 \approx \sum_{i=1}^{N} g'(\omega_{\mathrm{old}}^T x_i)^2 \left[\left(\omega_{\mathrm{old}}^T x_i + \frac{y_i - g(\omega_{\mathrm{old}}^T x_i)}{g'(\omega_{\mathrm{old}}^T x_i)}\right) - \omega^T x_i\right]^2. \tag{5}$$

- 最小化(5)的右边相当于做一个加权最小二乘
- 这个估计 g 再搜索 ω 的步骤不断循环直到收敛
- 若 $M > 1$，则用 $y - \sum_{i=1}^{m} g_m(\omega_m^T x)$ 后对 ω_{m+1} 和 g_{m+1} 进行上述估计和搜索
- 其实，M 本身也是搜索对象。我们通过不断评估拟合的优良程度，当发现精度提高小于预设的值时，M 的值就确定下来了。

算法

- 令 ω_{old} 为现在对 ω 的估计，

$$g(\omega^T x_i) \approx g(\omega_{\mathrm{old}}^T x_i) + g'(\omega_{\mathrm{old}}^T x_i)(\omega - \omega_{\mathrm{old}})^T x_i \tag{4}$$

- 代入(3)，得到

$$\sum_{i=1}^{N}\left[y_i - g(\omega^T x_i)\right]^2 \approx \sum_{i=1}^{N} g'(\omega_{\mathrm{old}}^T x_i)^2\left[\left(\omega_{\mathrm{old}}^T x_i + \frac{y_i - g(\omega_{\mathrm{old}}^T x_i)}{g'(\omega_{\mathrm{old}}^T x_i)}\right) - \omega^T x_i\right]^2. \tag{5}$$

- 最小化(5)的右边相当于做一个加权最小二乘
- 这个估计 g 再搜索 ω 的步骤不断循环直到收敛
- 若 $M > 1$，则用 $y - \sum_{i=1}^{m} g_m(\omega_m^T x)$ 后对 ω_{m+1} 和 g_{m+1} 进行上述估计和搜索
- 其实，M 本身也是搜索对象。我们通过不断评估拟合的优良程度，当发现精度提高小于预设的值时，M 的值就确定下来了。

算法

- 令 ω_{old} 为现在对 ω 的估计，

$$g(\omega^T x_i) \approx g(\omega_{\mathrm{old}}^T x_i) + g'(\omega_{\mathrm{old}}^T x_i)(\omega - \omega_{\mathrm{old}})^T x_i \tag{4}$$

- 代入(3)，得到

$$\sum_{i=1}^{N}\left[y_i - g(\omega^T x_i)\right]^2 \approx \sum_{i=1}^{N} g'(\omega_{\mathrm{old}}^T x_i)^2 \left[\left(\omega_{\mathrm{old}}^T x_i + \frac{y_i - g(\omega_{\mathrm{old}}^T x_i)}{g'(\omega_{\mathrm{old}}^T x_i)}\right) - \omega^T x_i\right]^2. \tag{5}$$

- 最小化(5)的右边相当于做一个加权最小二乘
- 这个估计 g 再搜索 ω 的步骤不断循环直到收敛
- 若 $M > 1$，则用 $y - \sum_{i=1}^{m} g_m(\omega_m^T x)$ 后对 ω_{m+1} 和 g_{m+1} 进行上述估计和搜索
- 其实，M 本身也是搜索对象。我们通过不断评估拟合的优良程度，当发现精度提高小于预设的值时，M 的值就确定下来了。

算法

- 令 ω_{old} 为现在对 ω 的估计，

$$\mathrm{g}(\omega^{\mathrm{T}}\mathrm{x_i}) \approx \mathrm{g}(\omega_{\mathrm{old}}^{\mathrm{T}}\mathrm{x_i}) + \mathrm{g}'(\omega_{\mathrm{old}}^{\mathrm{T}}\mathrm{x_i})(\omega - \omega_{\mathrm{old}})^{\mathrm{T}}\mathrm{x_i} \tag{4}$$

- 代入(3)，得到

$$\sum_{\mathrm{i}=1}^{\mathrm{N}}\left[\mathrm{y_i} - \mathrm{g}(\omega^{\mathrm{T}}\mathrm{x_i})\right]^2 \approx \sum_{\mathrm{i}=1}^{\mathrm{N}}\mathrm{g}'(\omega_{\mathrm{old}}^{\mathrm{T}}\mathrm{x_i})^2\left[\left(\omega_{\mathrm{old}}^{\mathrm{T}}\mathrm{x_i} + \frac{\mathrm{y_i} - \mathrm{g}(\omega_{\mathrm{old}}^{\mathrm{T}}\mathrm{x_i})}{\mathrm{g}'(\omega_{\mathrm{old}}^{\mathrm{T}}\mathrm{x_i})}\right) - \omega^{\mathrm{T}}\mathrm{x_i}\right]^2. \tag{5}$$

- 最小化(5)的右边相当于做一个加权最小二乘
- 这个估计 g 再搜索 ω 的步骤不断循环直到收敛
- 若 $\mathrm{M} > 1$，则用 $\mathrm{y} - \sum_{\mathrm{i}=1}^{\mathrm{m}}\mathrm{g_m}(\omega_{\mathrm{m}}^{\mathrm{T}}\mathrm{x})$ 后对 $\omega_{\mathrm{m+1}}$ 和 $\mathrm{g_{m+1}}$ 进行上述估计和搜索
- 其实，M 本身也是搜索对象。我们通过不断评估拟合的优良程度，当发现精度提高小于预设的值时，M 的值就确定下来了。

算法

- 令 ω_{old} 为现在对 ω 的估计，

$$g(\omega^T x_i) \approx g(\omega_{old}^T x_i) + g'(\omega_{old}^T x_i)(\omega - \omega_{old})^T x_i \tag{4}$$

- 代入(3)，得到

$$\sum_{i=1}^{N}\left[y_i - g(\omega^T x_i)\right]^2 \approx \sum_{i=1}^{N} g'(\omega_{old}^T x_i)^2 \left[\left(\omega_{old}^T x_i + \frac{y_i - g(\omega_{old}^T x_i)}{g'(\omega_{old}^T x_i)}\right) - \omega^T x_i\right]^2. \tag{5}$$

- 最小化(5)的右边相当于做一个加权最小二乘
- 这个估计 g 再搜索 ω 的步骤不断循环直到收敛
- 若 $M > 1$，则用 $y - \sum_{i=1}^{m} g_m(\omega_m^T x)$ 后对 ω_{m+1} 和 g_{m+1} 进行上述估计和搜索
- 其实，M 本身也是搜索对象。我们通过不断评估拟合的优良程度，当发现精度提高小于预设的值时，M 的值就确定下来了。

算法

- 令 ω_{old} 为现在对 ω 的估计，

$$g(\omega^T x_i) \approx g(\omega_{\mathrm{old}}^T x_i) + g'(\omega_{\mathrm{old}}^T x_i)(\omega - \omega_{\mathrm{old}})^T x_i \tag{4}$$

- 代入(3)，得到

$$\sum_{i=1}^{N}\left[y_i - g(\omega^T x_i)\right]^2 \approx \sum_{i=1}^{N} g'(\omega_{\mathrm{old}}^T x_i)^2\left[\left(\omega_{\mathrm{old}}^T x_i + \frac{y_i - g(\omega_{\mathrm{old}}^T x_i)}{g'(\omega_{\mathrm{old}}^T x_i)}\right) - \omega^T x_i\right]^2. \tag{5}$$

- 最小化(5)的右边相当于做一个加权最小二乘
- 这个估计 g 再搜索 ω 的步骤不断循环直到收敛
- 若 $M > 1$，则用 $y - \sum_{i=1}^{m} g_m(\omega_m^T x)$ 后对 ω_{m+1} 和 g_{m+1} 进行上述估计和搜索
- 其实，M 本身也是搜索对象。我们通过不断评估拟合的优良程度，当发现精度提高小于预设的值时，M 的值就确定下来了。

算法

1. 令 $j=1$，$m=1$，计算 $y_i^{(j)} = y_i - \bar{y}$，
2. 随机找一个 $\omega_m^{(j)}$ 作为尝试方向，计算 $v_{m,i}^{(j)} = \omega_m^{(j)T} x_i$
3. 用 $y_i^{(j)}$ 和 $v_{m,i}$ 估计 $g_m^{(j)}$
4. 计算 $\theta = \omega_m^{(j)T} x_i$，计算 $\xi = [g_m^{(j)'}(\omega_m^{(j)T} x_i)]$ 和 $\zeta = y_i^{(j)} - g_m^{(j)}(\omega_m^{(j)T} x_i)$
5. 以 ξ^2 为权重，将 $(\theta + \zeta/\xi)$ 对 x_i 做加权最小二乘，求得 ω_m，并记之为 $\omega_m^{(j+1)}$
6. $j \leftarrow j+1$，goto 3直到收敛
7. $m \leftarrow m+1, j=1$，goto 2直到

$$\sum_{i=1}^{N} (y_i - \sum_{m=1}^{M} g_m(v_{m,i}))^2 \tag{6}$$

变化不显著为止。

算法

1. 令 $j=1$，$m=1$，计算 $y_i^{(j)}=y_i-\bar{y}$，
2. 随机找一个 $\omega_m^{(j)}$ 作为尝试方向，计算 $v_{m,i}^{(j)}=\omega_m^{(j)T}x_i$
3. 用 $y_i^{(j)}$ 和 $v_{m,i}$ 估计 $g_m^{(j)}$
4. 计算 $\theta=\omega_m^{(j)T}x_i$，计算 $\xi=[g_m^{(j)'}(\omega_m^{(j)T}x_i)]$ 和 $\zeta=y_i^{(j)}-g_m^{(j)}(\omega_m^{(j)T}x_i)$
5. 以 ξ^2 为权重，将 $(\theta+\zeta/\xi)$ 对 x_i 做加权最小二乘，求得 ω_m，并记之为 $\omega_m^{(j+1)}$
6. $j\leftarrow j+1$，goto 3直到收敛
7. $m\leftarrow m+1, j=1$，goto 2直到

$$\sum_{i=1}^{N}(y_i-\sum_{m=1}^{M}g_m(v_{m,i}))^2 \tag{6}$$

变化不显著为止。

算法

1. 令 $j=1$，$m=1$，计算 $y_i^{(j)}=y_i-\bar{y}$，
2. 随机找一个 $\omega_m^{(j)}$ 作为尝试方向，计算 $v_{m,i}^{(j)}=\omega_m^{(j)T}x_i$
3. 用 $y_i^{(j)}$ 和 $v_{m,i}$ 估计 $g_m^{(j)}$
4. 计算 $\theta=\omega_m^{(j)T}x_i$，计算 $\xi=[g_m^{(j)'}(\omega_m^{(j)T}x_i)]$ 和 $\zeta=y_i^{(j)}-g_m^{(j)}(\omega_m^{(j)T}x_i)$
5. 以 ξ^2 为权重，将 $(\theta+\zeta/\xi)$ 对 x_i 做加权最小二乘，求得 ω_m，并记之为 $\omega_m^{(j+1)}$
6. $j\leftarrow j+1$，goto 3直到收敛
7. $m\leftarrow m+1, j=1$，goto 2直到

$$\sum_{i=1}^{N}(y_i-\sum_{m=1}^{M}g_m(v_{m,i}))^2 \tag{6}$$

变化不显著为止。

算法

1. 令 $j=1$，$m=1$，计算 $y_i^{(j)}=y_i-\bar{y}$，
2. 随机找一个 $\omega_m^{(j)}$ 作为尝试方向，计算 $v_{m,i}^{(j)}=\omega_m^{(j)T}x_i$
3. 用 $y_i^{(j)}$ 和 $v_{m,i}$ 估计 $g_m^{(j)}$
4. 计算 $\theta=\omega_m^{(j)T}x_i$，计算 $\xi=[g_m^{(j)'}(\omega_m^{(j)T}x_i)]$ 和 $\zeta=y_i^{(j)}-g_m^{(j)}(\omega_m^{(j)T}x_i)$
5. 以 ξ^2 为权重，将 $(\theta+\zeta/\xi)$ 对 x_i 做加权最小二乘，求得 ω_m，并记之为 $\omega_m^{(j+1)}$
6. $j\leftarrow j+1$，goto 3直到收敛
7. $m\leftarrow m+1, j=1$，goto 2直到

$$\sum_{i=1}^{N}(y_i-\sum_{m=1}^{M}g_m(v_{m,i}))^2 \tag{6}$$

变化不显著为止。

算法

1. 令 $j=1$，$m=1$，计算 $y_i^{(j)}=y_i-\bar{y}$，
2. 随机找一个 $\omega_m^{(j)}$ 作为尝试方向，计算 $v_{m,i}^{(j)}=\omega_m^{(j)T}x_i$
3. 用 $y_i^{(j)}$ 和 $v_{m,i}$ 估计 $g_m^{(j)}$
4. 计算 $\theta=\omega_m^{(j)T}x_i$，计算 $\xi=[g_m^{(j)'}(\omega_m^{(j)T}x_i)]$ 和 $\zeta=y_i^{(j)}-g_m^{(j)}(\omega_m^{(j)T}x_i)$
5. 以 ξ^2 为权重，将 $(\theta+\zeta/\xi)$ 对 x_i 做加权最小二乘，求得 ω_m，并记之为 $\omega_m^{(j+1)}$
6. $j\leftarrow j+1$，goto 3直到收敛
7. $m\leftarrow m+1, j=1$，goto 2直到

$$\sum_{i=1}^{N}(y_i-\sum_{m=1}^{M}g_m(v_{m,i}))^2 \tag{6}$$

变化不显著为止。

算法

1. 令 $j=1$，$m=1$，计算 $y_i^{(j)}=y_i-\bar{y}$，
2. 随机找一个 $\omega_m^{(j)}$ 作为尝试方向，计算 $v_{m,i}^{(j)}=\omega_m^{(j)T}x_i$
3. 用 $y_i^{(j)}$ 和 $v_{m,i}$ 估计 $g_m^{(j)}$
4. 计算 $\theta=\omega_m^{(j)T}x_i$，计算 $\xi=[g_m^{(j)\prime}(\omega_m^{(j)T}x_i)]$ 和 $\zeta=y_i^{(j)}-g_m^{(j)}(\omega_m^{(j)T}x_i)$
5. 以 ξ^2 为权重，将 $(\theta+\zeta/\xi)$ 对 x_i 做加权最小二乘，求得 ω_m，并记之为 $\omega_m^{(j+1)}$
6. $j\leftarrow j+1$，goto 3直到收敛
7. $m\leftarrow m+1, j=1$，goto 2直到

$$\sum_{i=1}^{N}(y_i-\sum_{m=1}^{M}g_m(v_{m,i}))^2 \tag{6}$$

变化不显著为止。

算法

1. 令 $j=1$，$m=1$，计算 $y_i^{(j)}=y_i-\bar{y}$，
2. 随机找一个 $\omega_m^{(j)}$ 作为尝试方向，计算 $v_{m,i}^{(j)}=\omega_m^{(j)T}x_i$
3. 用 $y_i^{(j)}$ 和 $v_{m,i}$ 估计 $g_m^{(j)}$
4. 计算 $\theta=\omega_m^{(j)T}x_i$，计算 $\xi=[g_m^{(j)'}(\omega_m^{(j)T}x_i)]$ 和 $\zeta=y_i^{(j)}-g_m^{(j)}(\omega_m^{(j)T}x_i)$
5. 以 ξ^2 为权重，将 $(\theta+\zeta/\xi)$ 对 x_i 做加权最小二乘，求得 ω_m，并记之为 $\omega_m^{(j+1)}$
6. $j\leftarrow j+1$，goto 3直到收敛
7. $m\leftarrow m+1, j=1$，goto 2直到

$$\sum_{i=1}^{N}(y_i-\sum_{m=1}^{M}g_m(v_{m,i}))^2 \tag{6}$$

变化不显著为止。