统计学习引论笔记

Kevin

目录

1 基本概念

假定我们观测到一个定量应变量 Y，p 个不同的自变量 $X = (X_1, \cdots, X_p)$，它们之间存在关系：

$$Y = f(X) + \epsilon.$$

其中，f 是某个固定但未知的函数，ϵ 是随机误差项（error term），ϵ 与 X 独立且均值为 0，f 表示 X 为 Y 提供的系统性信息。简单来说，统计学习就是一系列估计 f 的方法。

为什么要估计 f？主要有两种目的：预测和推断。从预测来说，我们获得 f 的估计 $\hat{f}$，然后从 X 可以估计 Y：$\hat{Y} = \hat{f}(X)$。$\hat{f}$ 一般被当成一个黑箱（black box），只要能通过它得到准确的估计，我们不太关心它的具体形式。$\hat{Y}$ 的准确性取决于两个误差：可消除误差和不可消除误差。统计学习方法的目标就是最小化可消除误差。从推断来说，我们需要考察自变量如何影响应变量，此时 $\hat{f}$ 不能看作黑箱了，我们需要知道它的准确形式。

怎么估计 $\hat{f}$？我们需要一系列训练数据：

$$\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}, \quad \text{where} \quad x_i = (x_{i1}, x_{i2}, \cdots, x_{ip})^T.$$

基于这些训练数据，我们需要寻找某个函数 $\hat{f}$，满足：

$$Y \approx \hat{f}(X), \forall (X, Y).$$

大致上，统计学习方法分为参数方法和非参数方法。

1.1 评估模型准确性

对于回归问题，最常用的测度是均方误（mean squared error，MSE）：

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{f}(x_i))^2.$$

2 线性回归

面对一个数据集，比如 Advertising，包括四个变量，sales，TV，radio，newspaper。我们要问几个问题：

1. 广告预算与销售额之间有关系吗?
2. 如果有上述关系，这种关系有多强?
3. 哪种广告媒介影响销售额?
4. 广告媒介对销售额的影响能准确估计吗?
5. 能准确预测未来的销售额吗?
6. 广告支出与销售额之间的关系是线性的吗?
7. 广告媒介之间存在协同效应吗?

2.1 简单线性回归

假设 X 和 Y 之间近似存在线性关系：

$$Y \approx \beta_0 + \beta_1 X.$$

一旦获得了参数的估计值 $\hat{\beta}_0, \hat{\beta}_1$，我们就可以根据某个特定的 X，预测 Y：

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x,$$

其中，$\hat{y}$ 表示在 $X = x$ 时 Y 的一个预测。

我们要找到最能够拟合数据集的参数估计，使得组成的直线尽可能靠近数据点。有多种方式来衡量 “近”，最常用的方法之一是最小二乘法（OLS）。简单线性回归的 OLS 估计为：

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2},$$
$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x},$$

其中，$\bar{y}, \bar{x}$ 为样本均值。

2.2 多元线性回归

有如下模型：

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon.$$

作多元回归，一般要回答四个问题：

1. 响应变量和自变量之间相关吗?
2. 是否所有的自变量都能解释 Y，抑或只有一部分有用?
3. 模型对数据的拟合程度如何?
4. 给定自变量值，如何预测响应变量的值? 预测的准确性如何?

2.2.1 问题 1：响应变量与自变量之间有关系吗？

对于这个问题，可以用 F 检验。原假设为所有的参数均为 0：

$$H_0 : \beta_1 = \beta_2 = \cdots = \beta_p = 0; \quad H_a : \exists j, \beta_j \neq 0.$$

构造 F 统计量：

$$F = \frac{(\text{TSS} - \text{RSS})/p}{\text{RSS}/(n-p-1)},$$

可以证明，当原假设为真时，上述统计量中分子和分母的期望值均为 σ^2（误差方差），因此，如果自变量和响应变量不存在关系，F 统计量应该接近 1。到底 F 多大时，可以拒绝原假设？这要看 n 和 p 的大小。当 n 较大时，即使 F 统计量只比 1 稍大也可以拒绝原假设。反之，当 n 比较小时，需要更大的 F 值。当 H_0 为真且误差项服从正态分布时，F 统计量服从 F 分布。由此，可以从相应的 p 值来判断是否可以拒绝原假设。

（TODO：F 检验与 t 检验的关系，为何需要 F 检验？）

2.2.2 问题 2：哪些变量是重要的？

多元回归分析的第一步，是计算 F 统计量并检查相应的 p 值。如果这一步得出至少有一个自变量与响应变量有关，那么很自然地，我们要找出哪些是重要的（应该进入模型），这项任务称为“变量选择”。我们可以查看每一个变量的 p 值（t 检验），但是如果自变量数量很多（p 很大），这种办法会导致错误。

通过选取不同的变量，可以生成 2^p 个模型。当 p 较大时，尝试所有的模型显然是不切实际的。我们需要一种自动且有效的方法来选取一部分模型进行考虑。有三种传统办法：前向选择、后向选择（当 $p > n$ 时不可用）和混合选择。

2.2.3 问题 3：模型拟合性

最常用的两个衡量拟合性的数值是 RSE 和 R^2。一元回归中，$R^2 = \text{Cov}(X,Y)^2$；在多元线性回归中，$R^2 = \text{Cov}(Y,\hat{Y})^2$。事实上，拟合线性模型的一个性质就是，使这个相关系数在所有线性模型中最大。

除了看 RSE 和 R 方，还可以通过图形来观察。在销售额的例子中，如果作出 $\text{sales} \sim \text{TV} + \text{radio}$ 的三维图，从残差的分布情况可以看到，变量之间存在非线性，意味着可能需要加入交互项（协同效应）。

2.2.4 问题 4：预测

利用回归得到的参数估计来预测，会面临三类不确定性：

1. 可减少误差。来自 OLS 回归平面 $\hat{Y}$ 与真实总体回归平面 $f(X)$ 之间的差距，可以由置信区间来表示。
2. 模型偏差。来自线性模型与实际不符产生的误差，这种误差也是可减少的。
3. 不可减少误差。即使我们知道真实的参数，也会面临由随机误差项带来的不可减少的误差。也就是 Y 与 $\hat{Y}$ 的差距。可以用预测区间来表示。

（TODO：预测区间与置信区间的区别?）

2.3 回归模型中的其他问题

2.3.1 定性预测子

定性变量一般称为因子（factor），其可能的取值称为水平（level）。对于仅有两个水平的因子，将其加入模型是非常简单的，只要构造一个哑变量。比如，对于 gender 变量，可以构造一个新的变量 x_i，当第 i 个人为女性时取 1，反之取 0。相应的回归模型可以写成：

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i = \begin{cases} \beta_0 + \beta_1 + \epsilon_i & 第i个为女性 \\ \beta_0 + \epsilon_i & 第i个为男性 \end{cases}$$

回归系数的解释：对于取三个可能值的变量，可以定义两个亚变量，模型参数的解释类似。定性变量的编码方式不唯一，但不同方式都将得到同样的模型拟合结果，只是系数及其解释不同（都表示某种比较）。

2.3.2 线性模型的拓展

标准线性回归模型带有很多假设，其中涉及自变量和应变量关系的两个重要假设是可加性（additive）和线性（linear）。

我们可以通过加入交互项放松可加性条件。以 Advertising 数据集为例，可加入 radio × TV。有时，加入的交互项显著，而主效应却不显著。此时，要不要包括主效应呢？统计学习中有如下原理。

层次原理（hierarchical principle）：如果模型中包括交互项，即使主效应对应的 p 值不显著，也必须包括在模型中。

交互项可以包含定量和定性的变量。比如 Credit 数据集，设想我们希望通过 income 和 student 预测 balance。当不包括交互项时，模型可以表

示为两条平行线，income 对 balance 的平均影响与是不是学生无关。这样的模型是有局限的，解决办法是加入交互项。加入收入和学生的交互项后，可以得到两条回归线，截距和斜率均不同。

体现非线性关系的一种简单形式是多项式回归。

2.3.3 潜在问题

拟合线性回归模型时常常会碰到如下问题：

1. 非线性关系。
2. 误差项相关。
3. 异方差。
4. 异常值。
5. 高杠杆值。
6. 共线性。

非线性关系。用残差对于拟合值作残差图，如果模型是线性的，残差图不应该呈现明显的模式。如果残差图显示有非线性特征，可以对自变量进行简单非线性变换，比如 $\log X$，$\sqrt{X}$ 或 X^2 等。当然，还有更加高级的处理非线性问题的方法。

误差项相关。如果误差项相关，那么标准误差将被低估，带来的结果是，置信区间和预测区间都将变窄。比如，95% 的置信区间实际包含真实参数值的概率要比 0.95 小得多。另外，模型对应的 p 值也将变小，这将导致误以为参数是显著的。最常见的例子是时间序列。

异方差。当误差项的方差不同时，标准误、置信区间和相应的假设检验都将不靠谱。鉴别异方差的一种方法是看残差图是不是出现所谓的漏斗状（funnel shape）。出现这种情况，一种解决办法是对应变量 Y 进行对数或开方等非线性变换，从而大大缩小 Y 的值并减少异方差。有时，我们对每一个应变量的方差有较为明确的认识，这时可以采用某种形式的加权最小二乘法（WLS）。

异常值。所谓异常值，是指 y_i 与其预测值差距很大的数据点。异常值一般对模型拟合影响不大，但会对 RSE、R 方产生影响。残差图可以探测异常值，但实践中通常残差大小难以决断，这时可以用 t 化残差来判断，一般 t 化残差绝对值大于 3 的观测可能是异常值。模型分析中是否要移除异常值，需要谨慎考虑。

高杠杆值。与异常值关注 y_i 不同，高杠杆值是看 x_i 是否不同寻常。如果一个观测的自变量比其他观测明显要大时，可能就是高杠杆值。相比异常值，高杠杆值对模型拟合有较大影响。衡量一个观测的杠杆度，一般用杠杆统计量，对于简单线性回归，杠杆统计量由下式计算：

$$h_i = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{i'=1}^{n}(x_{i'} - \bar{x})^2}.$$

显然，如果 x_i 与 $\bar{x}$ 差异越大，h_i 越大。此外，杠杆值必定位于 $1/n$ 和 1 之间，所有观测的平均杠杆值等于 $(p+1)/n$。因此，当某个观测的杠杆值超过 $(p+1)/n$ 较多时，我们有理由怀疑它是高杠杆值。如果一个观测既是异常值，又是高杠杆值，是非常危险的组合，需要高度重视！

共线性。如果模型中某些自变量之间相关时，回归系数的准确性会降低，对应的标准误会增大，从而 t 统计量会减小，结果导致该拒绝原假设时没拒绝，也就是降低了假设检验的 power。一个检测共线性的简单办法是看自变量的相关矩阵，其中一个较大的值表示变量之间相关性较高。但是，相关矩阵只能看两个变量的关系，而共线性可能存在与三个及以上变量之间，也就是多重共线性。此时，更好的办法是计算方差膨胀因子（VIF）。计算如下：

$$\mathrm{VIF}(\hat{\beta}_j) = \frac{1}{1 - R^2_{X_j|X_{-j}}}.$$

其中，$R^2_{X_j|X_{-j}}$ 是 X_j 对其他所有自变量进行回归得到的 R 方。如果 $R^2_{X_j|X_{-j}}$ 接近 1，存在共线性，此时 VIF 很大。面临共线性问题，有两种常用解决办法，一种是去掉一个问题变量，另一种是将共线性的变量合并成一个变量（比如标准化后取平均）。

2.4 市场营销方案

现在可以用多元回归分析的结果回答开头提出的七个问题。

2.5 线性回归与 K-近邻

线性回归属于参数方法，优点易于拟合和解释，缺点是容易脱离实际，导致预测不准。非参数方法也可用于回归，一个简单著名的例子是 KNN 回归。KNN 回归与 KNN 分类器密切相关。给定 K 和预测点 x_0，KNN 回归首先找到与 x_0 最接近的 K 个训练点，组成集合 $\mathbb{N}_0$，接着，用这些训练点

的响应值的均值来估计 $f(x_0)$:

$$\hat{f}(x_0) = \frac{1}{K} \sum_{x_i \in \mathbb{N}_0} y_i.$$

选择多大的 K, 取决于偏差与方差之间的权衡。K 越小, 模型越 flexible, 偏差越小, 但同时方差也越大; 反之则反之。后面会讨论一些计算测试误差率的方法, 这些方法可用于 K 的选择。

到底是选择参数化的线性模型还是非参数方法（比如 KNN）呢? 一个简单的原则是, 当实际模型接近线性时, 就采用线性模型。可以证明, 对于变量个数较少的情况, 当模型的非线性程度增加时, 非参数方法的 test MSE 只有微小变化, 而参数化线性模型的 test MSE 却大大增加。换句话说, 当真实关系为线性时, KNN 比线性模型稍差; 但当真实关系为非线性时, KNN 却比线性模型好很多。这个特征会导致一种倾向, 就是 "KNN 应该优先选用", 然而, 实际上这并不对。主要问题在于, 当模型维数（自变量个数）增加时, 即使真实关系为非线性, KNN 的表现也会比线性模型糟糕得多。原因就是著名的 "**维数灾难**"。事实上, 即使在 p 较小时, 我们一般也会由于便于解释等原因而倾向于选用线性模型（愿意牺牲一点预测准确性)。

3 分类

当响应变量为定性（类别）变量时, 线性回归就不适用了。此时要用专门的分类方法。

3.1 Logistic 回归

考虑 Default 数据集, 其中, 应变量 default 表示两个类别: Yes 或者 No。Logistic 回归对 Y 属于某个类别的概率进行建模。比如, 给定 balance 的值, default 的概率表示为 $\Pr(\text{default} = \text{Yes} \mid \text{balance})$, 简记为 $p(\text{balance})$。为了方便, 我们采用 0/1 编码, 也就是说我们想考察 $p(X) = \Pr(Y = 1 \mid X)$ 与 X 之间的关系。为了确保概率值位于 0 和 1 之间, 我们取如下 **logistic 函数**:

$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}.$$

稍加变换:

$$\frac{p(X)}{1 - p(X)} = e^{\beta_0 + \beta_1 X}.$$

其中，左边称为**几率**（odds）。再取对数：

$$\log(\frac{p(X)}{1-p(X)}) = \beta_0 + \beta_1 X.$$

左边称为**对数几率**（log-odds，简称 logit）。可以看到，logit 是 X 的线性函数。

logistic 回归的参数估计一般采用最大似然法（maximum likelihood），选择系数 β_0, β_1，使下面的似然函数最大：

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'})).$$

与线性回归的 t-统计量类似，logistic 回归计算 z-统计量（系数估计与其标准误之比），如果 z 的绝对值较大，拒绝 H_0。回归中的截距项主要是用来规范概率值的，一般不作考虑。一旦系数估计出来，就可用于预测。

多元 logistic 回归过程类似。然而，需要注意 **confounding** 现象。

3.2 线性判别分析

概括而言，logistic 回归是**直接**对条件概率进行建模。而判别分析是一种**间接**的方法，主要基于 Bayes 原理。具体来说，先对 Y 取不同值条件下 X 的分布建模，然后使用 Bayes 定理反转概率。如果 X 的条件分布是正态的，可以证明此时的模型与 logistic 回归很接近。

（TODO：为什么需要判别分析，它相对于 logistic 回归有什么好处?）

假定有 K 个可能类别。令 π_k 表示一个观测来自类别 k 的总体（overall）概率，一般称这个概率为先验概率（prior）。令 $f_k(X) \equiv \Pr(X = x \mid Y = k)$ 表示来自 k 类别的观测的密度函数。它衡量的是，某个属于 k 类的观测之自变量值接近 x 的概率。根据 Bayes 定理：

$$\Pr(Y = k \mid X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}.$$

与前面的记法保持一致，我们一般将上面的条件概率写成 $p_k(X)$ 的形式。接下来要做的是，估计 π_k 和 $f_k(X)$，然后将它们代入上面的公式。当我们拥有 Y 的一个随机样本，计算 π_k 很容易，只要基于训练集计算相应的比例即可。然而，估计 $f_k(X)$ 可没这么简单了，除非我们对那些密度函数的形式做些简单的假定。一般称 $p_k(x)$ 为一个满足 $X = x$ 的观测属于类别 k 的后

验概率（posterior）。我们知道，Bayes 分类器具有最低的错误率。因此，如果能找到一种方法估计出 $f_k(X)$，我们就拥有一个近似于 Bayes 的分类器。

先看 $p=1$ 的情况。为了估计 f_k，我们需要对其形式做些假定。比如，假定 f_k 服从正态分布，μ_k, σ_k^2 分别为均值和方差。先假定所有类别具有等方差，即 $\sigma_1^2 = \cdots = \sigma_K^2 = \sigma^2$，从而，我们写出此时的后验概率：

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_k)^2\right)}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_l)^2\right)}.$$

取对数，简单变换后，可以证明，上式最大等价于下式最大：

$$\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k).$$

比如，当 $K=2, \pi_1 = \pi_2$ 时，Bayes 决策边界对应于点：

$$x = \frac{\mu_1^2 - \mu_2^2}{2(\mu_1 - \mu_2)} = \frac{\mu_1 + \mu_2}{2}.$$

实际中，即使我们对正态假设非常肯定，仍需对参数 μ, π, σ^2 进行估计。我们采用如下的估计：

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

$$\hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2$$

其中，n 为观测总数，n_k 为第 k 类中的观测数，$\hat{\sigma}^2$ 可以看作 K 个类别的样本方差的加权平均。有时我们知道关于 π_k 的信息，当不知道这些信息时，LDA 通过相应的观测占比来估计，亦即 $\hat{\pi}_k = n_k/n$。有了这些估计，我们就可以对某个观测进行分类，目标是使得下式最大：

$$\hat{\delta}_k(x) = x \cdot \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log(\hat{\pi}_k)$$

上式称为判别函数（discriminant function），可见它是线性的。

当 $p>1$ 时，LDA 假定第 k 类中的观测来自于多元正态分布 $N(\mu_k, \mathbf{\Sigma})$，其中，$\mu_k$ 为 k 类的均值向量，$\mathbf{\Sigma}$ 为（共同的）协方差阵。判别函数为：

$$\delta_k(x) = x^T \mathbf{\Sigma}^{-1} \mu_k - \frac{1}{2} \mu_k^T \mathbf{\Sigma}^{-1} \mu_k + \log \pi_k.$$

Bayes 决策边界为 $\{x : \delta_k(x) = \delta_l(x), k \neq l\}$。实际中，与一元情形类似，我们也需要估计那些未知参数。

举个例子，对 Default 数据集进行线性判别分析，根据某个人的信用卡透支额和是不是学生来预测他会不会违约。对 10000 个训练数据应用 LDA 模型拟合后，训练误差率为 2.75%。对于这个看似较低的误差率，我们要注意两点：

1. 训练误差率往往比测试误差率小，但恰恰后者才是我们主要关心的东西。特别是当参数个数 p 相对于样本数 n 较大时，可能产生过拟合（overfitting）。
2. 在这个栗子中，实际上样本中本来就只有 3.33% 的人违约，因此，我们不凭任何信息，瞎猜一个人总不违约，误差率也就是 3.33%。也就是说，一个 null 分类器也仅仅比 LDA 误差率高一点点。

实际中，像上面这个二元分类器会犯两类错误：将违约的人分到 no default 类别，或者将不违约的人分到 default 类别。一般我们用 confusion matrix 来表示这两类错误。对上面的例子，LDA 预测共有 104 个人违约，其中 81 个真的违约而 23 个并没有。因此，9667 个未违约的人中只有 23 个没有被正确预测，这个错误率似乎非常低！但是，333 个违约的人中，252 个人（75.7%）未被预测到。也就是说，尽管总体错误率比较低，对于违约的人的预测错误率却很高。信用卡公司一般致力于找出高风险的客户，75.7% 的错误率是不能接受的。医学和生物学中一般用敏感性（sensitivity）和设定率（specificity）来衡量模型的表现。在上面的例子中，敏感性表示真实违约者被识别的比例（24.3%），设定率表示未违约者被正确识别的比例（99.8%）。

（TODO：为什么信用卡违约例子中 LDA 的表现这么差（敏感性这么低）？——关键在于临界值的选取。）

ROC 曲线可以用来同时展示各种临界值下的两类错误率，即 True positive rate 和 False positive rate。

LDA 假定所有类别具有相同的协方差阵，二次判别分析（QDA）假定协方差阵不同，即假定来自类别 k 的观测 $X \sim N(\mu_k, \mathbf{\Sigma}_k)$。此时，判别函数是 x 的二次函数。什么时候选用 LDA 或是 QDA 呢？答案是偏差 - 方差权衡。

4 重抽样方法

最常用的重抽样方法包括交叉验证（cross-validation）和自举（bootstrap）。交叉验证常用于模型评估（model assessment）和模型选择

(model selection)。

4.1 交叉验证

LOOCV 由于要拟合 n 次，如果 n 较大，原则上计算非常耗时。幸好 LOOCV 有个好的性质：当采用最小二乘拟合线性或多项式模型时，LOOCV 的成本相当于拟合模型 1 次！有下式成立：

$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - h_i}\right)^2,$$

其中，$\hat{y}_i$ 是原始最小二乘拟合值，h_i 是杠杆值。

LOOCV 是 k-fold CV 的特例，其中 $k = n$。实际中人们一般用 $k = 5$ 或者 $k = 10$，最明显的好处是计算量，其他的好处包括偏差-方差权衡等等。我们进行交叉验证，有时只是关心估计 test-MSE 曲线最低点的位置而不是具体的值，因为我们想要比较不同统计学习方法或者不同弹性，看谁的测试误差最低。

除了计算量上的好处外，k-fold CV 相比 LOOCV，对测试误差的估计更加准确，这与偏差-方差权衡有关。从减少偏差的角度看，LOOCV 优于 k-fold CV。但是，从方差角度看，当 $k < n$ 时，LOOCV 的方差要高于 k-fold CV。为何？在 LOOCV 中，我们实际上在对 n 个拟合模型作平均，其中每一个模型的训练集几乎相同，因此这些结果之间高度（正）相关。与此不同，k-fold CV，我们对 $k < n$ 个相关性较低的拟合模型作平均。因为很多高度相关的量的均值具有较大的方差，所以 LOOCV 得到的测试误差估计具有较高的方差。经验研究表明，$k = 5$ 或 10 的 k-fold CV 既不会有太大的偏差也不会有很高的方差。

当应变量是分类变量时，误差率计算主要基于误分类观测数，比如对于 LOOCV，误差率公式为：

$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\mathrm{Err}_i,$$

其中，$\mathrm{Err}_i = I(y_i \neq \hat{y}_i)$。

4.2 Bootstrap

自举（bootstrap）用来量化估计量或者统计学习方法不确定性。自举的实质是可放回重复抽样（sampling with replacement）。

5 线性模型选择与正则化

由于预测准确性和模型解释性等原因，我们需要比 OLS 更好的拟合方法。这里我们讨论三类重要的方法：子集选择（subset selection）、收缩（shrinkage）和降维（dimension reduction）。

5.1 子集选择

子集选择的方法包括最佳子集（best subset）法和逐步选择法。

算法：最佳子集选择

1. Let M_0 denote the null model, which contains no predictors. This model simply predicts the sample mean for each observation.
2. For $k = 1, 2, \cdots, p$:

a. Fit all $\binom{p}{k}$ models that contain exactly k predictors.
b. Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here best is defined as having the smallest RSS, or equivalently largest R^2.

3. Select a single best model from among $\mathcal{M}_0, \cdots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

算法：前向选择法

算法：后向选择法

前面这些方法中都生成一系列包含自变量某个子集的模型，如何从中选取最优模型呢？由于我们关心的是测试误差，所以传统的 RSS 或者 R-方都不能用作判断指标，这些只适用于训练误差。要估计测试误差，有两种方法：

1. 间接法：对训练误差作某种调整（adjustment），消除由于过拟合导致的偏差。
2. 直接法：采用验证集或交叉验证法直接估计测试误差。

用来调整训练误差的指标包括 C_p，AIC，BIC，调整 R 方等。

C_p——对于包括 d 个自变量的最小二乘拟合模型，测试 MSE 的 C_p 估计为：

$$C_p = \frac{1}{n}\left(\mathrm{RSS} + 2d\hat{\sigma}^2\right),$$

其中，$\hat{\sigma}^2$ 是误差方差 ϵ 的估计。C_p 统计量实际上是在训练 RSS 上添加了一个惩罚项，用以调整训练误差对测试误差的低估。显然，惩罚项随着自变量数增加而增大，旨在抵消训练 RSS 的减少。可以证明，如果 $\hat{\sigma}^2$ 是 σ^2 的无偏估计，那么 C_p 就是测试 MSE 的无偏估计。因此，如果模型的测试误差较小，则 C_p 值较小，也就是说，我们选择最优模型，就是选择 C_p 最小的模型。

AIC——AIC 是针对采用最大似然估计的模型定义的。对具有高斯误差的多元线性回归来说，

$$\mathrm{AIC} = \frac{1}{n\hat{\sigma}^2}\left(\mathrm{RSS} + 2d\hat{\sigma}^2\right),$$

其中，为了方便，我们去除了可加的常数。对于 LS 模型来说，AIC 和 C_p 成比例。

BIC——BIC 是从 Bayes 角度来定义的，不过最后与 C_p (AIC) 也差不多：

$$\mathrm{BIC} = \frac{1}{n}\left(\mathrm{RSS} + \log(n)d\hat{\sigma}^2\right).$$

其中，也省略了常数。与 C_p 类似，我们选择 BIC 最低的模型。注意到，当 $n > 7$ 时 $\log n > 2$，因此，相比 C_p，BIC 对变量数较多的模型赋予了更大的惩罚。

调整 R 方——调整 R 方旨在解决 RSS 随变量数增加而减少的问题。

$$\text{调整}R^2 = 1 - \frac{\mathrm{RSS}/(n-d-1)}{\mathrm{TSS}/(n-1)}.$$

与前面三个标准不同，较大的调整 R 方意味着较小的测试误差。理论上，具有最大调整 R 方的模型仅包含应包含的变量（无噪声）。

与这些间接调整指标相比，用验证集或交叉验证方法直接估计测试误差更好，一来不用对真实模型作太多假设，二来应用范围更广（比如当无法确定自由度或者误差方差难以估计时）。在计算能力大增的今天，交叉验证方法显然更具吸引力。

当若干模型差不多好时，我们选择最简单的（包含最少的自变量）。一般称之为 **one-standard-error rule**：对不同 size 的模型，分别计算估计测试误差的标准误，然后从估计测试误差在最低点 one-standard-error 范围内的模型中选择最小的模型。

5.2 收缩法

与子集选择法选择部分自变量进行拟合不同，收缩法对包含所有自变量的模型进行拟合，同时采用某种技术对系数向 0 进行压缩（正则化）。最著名的两种压缩方法是岭回归（ridge regression）和 lasso。

岭回归和 lasso 都是在最小二乘模型上添加压缩惩罚项（shrinkage penalty），形成新的最小化目标函数。惩罚项由一个调节参数（tuning parameter）λ 和系数向量的某个范数的乘积构成，岭回归使用 ℓ_2 范数，lasso 使用 ℓ_1 范数。

虽然岭回归和 lasso 都将系数往 0 压缩，但 lasso 采用的 ℓ_1 范数能够强制某些系数正好等于 0（当调节系数足够大）。因此，lasso 实施了变量选择，出来的模型也比岭回归更容易解释。有时称 lasso 得到的模型为稀疏（sparse）模型。

可以换一种角度看两种压缩方法，即某种约束条件下的 RSS 最小化：

$$\text{lasso}: \min_{\beta} \ \text{RSS} \quad s.t. \sum_{j=1}^{p} |\beta_j| \leq s$$

$$\text{岭回归}: \min_{\beta} \ \text{RSS} \quad s.t. \sum_{j=1}^{p} \beta_j^2 \leq s$$

对于 $p = 2$ 的情形，lasso 的约束条件是一个菱形，岭回归的约束条件是一个圆。基于这种思路，最佳子集选择也可以重述为以下约束优化问题：

$$\min_{\beta} \ \text{RSS} \quad s.t. \sum_{j=1}^{p} I\left(\beta_j \neq 0\right) \leq s.$$

可以看到，最佳子集选择所述的最优化问题很难求解，而 lasso 和岭回归可以看成这一问题比较容易处理的版本，当然 lasso 要更像最佳子集选择一点。

到底 lasso 和岭回归哪个更好呢？具体问题具体分析。理论上说，可以通过自变量的重要性来判断，但是实际中与应变量相关的自变量个数事先未知，因此，需要通过交叉验证等方法来判断哪种方法更适用。

（TODO：岭回归和 lasso 的 Bayes 解释）

（TODO：调节系数的选择）

5.3 降维方法

前面这些用来降低方差的方法均使用原始的自变量，降维方法旨在对原始自变量进行某种变换后再拟合。令 $Z_1, Z_2, \cdots, Z_M$ 表示原来的 p 个自

变量的 $M < p$ 个线性组合，即：

$$Z_m = \sum_{j=1}^{p} \phi_{jm} X_j$$

其中，$\phi_{1m}, \cdots, \phi_{pm}$ 为常数，$m = 1, \cdots, M$。然后，拟合以下线性回归模型：

$$y_i = \theta_0 + \sum_{m=1}^{M} \theta_m z_{im} + \epsilon_i, \quad i = 1, \cdots, n$$

从而，模型的维数从原来的 $p+1$ 降到了 $M+1$。注意到，

$$\sum_{m=1}^{M} \theta_m z_{im} = \sum_{m=1}^{M} \theta_m \sum_{j=1}^{p} \phi_{jm} x_{ij} = \sum_{j=1}^{p} \sum_{m=1}^{M} \theta_m \phi_{jm} x_{ij} = \sum_{j=1}^{p} \beta_j x_{ij},$$

其中，

$$\beta_j = \sum_{m=1}^{M} \theta_m \phi_{jm}.$$

因此，降维后的模型可以看成原来线性回归模型的一个特例。降维对估计系数施加了约束，要求它们必须具备上式中线性组合的形式。这种约束可能会导致估计系数出现偏差。然而，当 p 比 n 大时，选择 $M \ll p$ 可以显著减少拟合系数的方差。

降维方法一般包括两个步骤：第一，获得变换后的自变量 $Z_1, \cdots, Z_M$；第二，用这 M 个变量拟合模型。但是，如何选择这 M 个变量（等价地，选取 ϕ_{jm}）可以有多种方式，比如主成分回归（PCR）和偏最小二乘法（PLS）。

5.3.1 主成分回归

主成分回归（PCR）旨在构造头 M 个主成分 $Z_1, \cdots, Z_M$ 用以拟合，主要思想是，通常一小部分变量就足够解释数据中大部分的变异性以及变量之间的关系。换句话说，我们假定 $X_1, \cdots, X_p$ 展现最多变异的方向就是与 Y 关联的方向。尽管这个假设不一定正确，但是事实证明这个近似足够给出好的结果。即使 PCR 基于 $M \ll p$ 个自变量进行回归的方式更为简单，但是它并不进行变量（特征）选择，也就是不能把问题落脚到少数几个变量上。从这个意义上，在岭回归和 lasso 之间，PCR 更像前者。事实上，岭回归可以看成 PCR 的一个连续版本（详见 ESL 第 3.5 节）。PCR 中主成分的个数 M 一般由交叉验证来确定。实际运用中，在生成主成分之前，我们通常对自变量进行标准化。

5.3.2 偏最小二乘法

与 PCR 类似，PLS 也是先找到原有特征的一系列线性组合，形成新的变量集用以拟合。但是，与 PCR 不同的是，PLS 在选取新特征时采用监督的方式，即利用应变量 Y 来识别新的特征，这些新的特征不仅能很好近似老的变量，还与应变量有关。

如何生成主成分? 先看第一个主成分。将 p 个自变量标准化，令 ϕ_{j1} 等于简单线性回归 $Y \sim X_j$ 得到的系数，这样就得到 Z_1。这个系数实际上与 Y 与 X_j 的相关系数成比例，从而，在计算 $Z_1 = \sum_{j=1}^{p} \phi_{j1} X_j$ 时，与应变量相关度最高的变量被赋予最大的权重。为了得到第二个主成分，先对变量按 Z_1 作调整，也就是将所有变量对 Z_1 作回归，取其残差，这些残差可以解释为第一主成分未能解释的信息。然后，利用这些正交后的数据，按照与计算 Z_1 同样的方法计算 Z_2。如此迭代下去，获得 PLS 的 M 个成分。最后，我们基于新的 M 个变量，按照 PCR 同样的方式拟合模型。与 PCR 类似，M 的大小也一般由交叉验证来确定。

5.4 关于“高维”的思考

传统统计学往往是在低维（$n \gg p$）环境下设计出来的。最近二十年来，新技术改变了金融、营销和医药领域的数据搜集方式，现在获取到包括无数特征的数据是很平常的事。一方面，p 可以非常大，另一方面由于成本、样本可得性或其他原因，我们能利用的观测数又是有限的。此时，很多传统方法，比如 OLS、logistic 回归、LDA 等，都不能用了。

6 线性模型的拓展

看一些简单的扩展，如多项式回归、阶梯函数、回归样条、光滑样条、局部回归和广义加法模型。这些扩展放松了线性假设，同时也尽可能保持了模型的解释性。

6.1 多项式回归

将标准线性模型替换为一个多项式函数:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_d x_i^d + \epsilon_i,$$

其中，ϵ_i 为误差项。对于足够大的阶数 d，多项式回归可以生成高度非线性的曲线。上式可以看成是 x_i 的各阶项的线性模型，因此可以用最小二乘来拟合。一般我们不使用 d 超过 3 或 4 的项，因为阶数过高的多项式曲线会变得过度灵活，形状也会很奇怪（当自变量靠近边界值的时候，尤其如此）。

对于分类变量，我们可以类似进行 logistic 回归：

$$\Pr(y_i > a|x_i) = \frac{\exp\left(\beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_d x_i^d\right)}{1 + \exp\left(\beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_d x_i^d\right)}.$$

6.2 阶梯函数

多项式回归对 X 的结构作出了全局（global）限定。与此不同，阶梯函数避免全局限定，将 X 取值范围划分了多个区间（bins），在各个区间分别拟合不同的常数。这相当于把一个连续变量转换为一个有序分类变量。具体来说，构造 K 个切点 $c_1, c_2, \cdots, c_K$，将 X 转换为 $K+1$ 个新的变量：

$$\begin{aligned}
C_0(X) &= I(X < c_1),\\
C_1(X) &= I(c_1 \le X < c_2),\\
C_2(X) &= I(c_2 \le X < c_3),\\
&\vdots\\
C_{K-1}(X) &= I(c_{K-1} \le X < c_K),\\
C_K(X) &= I(c_K \le X),
\end{aligned}$$

注意到，对于任意的 X，$C_0(X) + C_1(X) + \cdots + C_K(X) = 1$，因为 X 必定属于且只属于其中一个区间。用最小二乘法拟合如下模型：

$$y_i = \beta_0 + \beta_1 C_1(x_i) + \beta_2 C_2(x_i) + \cdots + \beta_K C_K(x_i) + \epsilon_i$$

对于给定的 X，$C_1, \cdots, C_K$ 中最多只有一个非零。当 $X < c_1$ 时，模型中所有的自变量均为 0，因此 β_0 就可以解释为 $X < c_1$ 时 Y 的均值。进一步，$\beta_j (j = 1, \cdots, K)$ 解释为 $c_j \le X < c_{j+1}$ 相对于 $X < c_1$ 时响应变量的平均增量。这类模型也叫分段常数回归（piecewise-constant regression）。

6.3 基函数

多项式回归和分段常数模型都是基函数（basis function）模型的特例。考虑可应用于 X 的一族函数或变换 $b_j(X), j = 1, \cdots, K$。拟合下述模型：

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots + \beta_K b_K(x_i) + \epsilon_i.$$

其中，基函数 $b_1(\cdot), \cdots, b_K(\cdot)$ 固定且已知。多项式回归就相当于取 $b_j(x_i) = x_i^j$，分段常数模型相当于取 $b_j(x_i) = I(c_j \leq x_i < c_{j+1})$。除了这两种形式，基函数还可以取其他多种形式，包括小波和傅里叶级数等等，回归样条就是其中一类非常灵活的方法。

6.4 回归样条

我们可以对 X 的不同取值区域作分段多项式回归（piecewise polynomial regression），每一段具有不同的系数。系数改变的点称为结点（knots）。如果我们设置 K 个结点，则需要拟合 $K+1$ 个多项式回归模型。线性模型为一次多项式模型，分段常数模型则是零次多项式。

分段回归可能在断点出现跳跃，需要对其进行约束，使拟合后的曲线连续乃至光滑。每加一个限制，就会释放一个自由度，从而降低拟合复杂度。比如，对分成 2 段（一个结点）的三次模型，默认有 8 个参数，自由度为 8，当加上“连续、一阶导数连续、二阶导数连续”三个条件后，自由度降为 5。称如此拟合的曲线为三次样条（cubic splines）。带 K 个结点的三次样条自由度通常为 $4 + K$。自由度为 d 的样条的一般定义为，它是 d 阶分段多项式回归，并且在每个结点从 0 到 $d-1$ 阶导数均连续。

利用基函数模型，可以将带 K 个结点的三次回归样条表示为：

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots + \beta_{K+3} b_{K+3}(x_i) + \epsilon_i,$$

上式中的基函数可采用多种形式。一种直接的方法是，从三次多项式 (x, x^2, x^3) 开始，然后对每一个结点都加上一个截断幂基（truncated power basis）。截断幂基函数定义为：

$$h(x, \xi) = (x - \xi)_+^3 = \begin{cases} (x-\xi)^3 & x > \xi \\ 0 \end{cases}$$

其中，xi 为结点。换句话说，拟合包含 K 个结点的三次样条，就是作包含截距项和 $3 + K$ 个自变量的最小二乘回归，自变量形式为 $X, X^2, X^3, h(X, \xi_1), h(X, \xi_2), \cdots, h(X, \xi_K)$，共要估计 $K + 4$ 个参数。

不幸的是，样条在自变量边缘值附近会出现高方差。可以对回归样条施加边界约束，比如要求边界处呈线性，此时的回归就称为自然样条（natural spline），出来的系数估计更加稳定，置信区间较窄。

那么，如何确定结点的个数和位置呢？实际运用中，一般先规定想要的自由度，然后由软件自动（均匀）分配。一种办法是，对不同结点数（自由度）进行尝试，选择曲线形态最好的情况。另一种更为客观的办法是采用交叉验证。

回归样条一般能给出比多项式回归更好的结果，因为回归样条没必要非得加入高阶项，只有通过增加结点数即可增加模型的灵活性。一般来说，回归样条生成更加稳定的估计。样条允许我们在函数变化较快的地方加入较多的结点，而在函数变化较慢的地方加入较少的结点。

6.5 光滑样条

对一个数据集进行曲线拟合，就是找到一个尽量光滑的函数 g，使得 RSS 尽量小。怎样保证 g 光滑？一个自然的办法是最小化下面的函数：

$$\sum_{i=1}^{n}\left(y_i - g(x_i)\right)^2 + \lambda \int g''(t)^2 dt$$

其中，λ 是非负调节参数。最小化上式的函数 g 称为光滑样条（smoothing spline）。可以看出，这个函数的构成类似于岭回归和 lasso，都是 “Loss+Penalty” 的形式。二阶导数衡量函数的 “粗糙度”，二阶导数 $g''(t)$ 绝对值较大，意味着 $g(t)$ 在 t 附近起伏较大。因此，上式中的积分可以看成函数 g 一阶导数变化情况的汇总。如果 g 很光滑，则 $g'(t)$ 近似为常数，从而上面的积分值应该比较小。另外，λ 越大，g 越光滑。当 $\lambda = 0$，惩罚项没有效果，g 将恰好内插所有观测。当 $\lambda \to \infty$，g 变成一条尽可能接近观测点的直线，事实上此时就是线性最小二乘。当 λ 取中间值时，g 逼近训练数据且具有一定的光滑性。可见，λ 控制着偏差-方差权衡。

$g(x)$ 具有一些特殊的性质：（1）它是一个分段三次多项式，结点在唯一值 $x_1, \cdots, x_n$，且结点处的一阶和二阶导数均连续；（2）在极端结点之外是线性的，换句话说，$g(x)$ 是具有结点 $x_1, \cdots, x_n$ 的自然三次样条！然而，它与前面按照基函数方法得到自然三次样条是不同的。事实上，它是一个缩减的版本，其中 λ 就用来控制缩减程度，也就控制了有效自由度（effective degrees of freedom）。可以证明，当 λ 从 0 增加到 ∞，有效自由度 df_λ 从 n 降为 2。

作光滑样条拟合时，我们不需要选择结点的位置，但是需要选择 λ。很自然的，我们用交叉验证的方法来选择。LOOCV 对光滑样条能高效地计算，基本上相当于拟合一次，我们用下面的公式：

$$\mathrm{RSS}_{cv}(\lambda)=\sum_{i=1}^{n}\left(y_i-\hat{g}_{\lambda}^{(-i)}(x_i)\right)^2=\sum_{i=1}^{n}\left[\frac{y_i-\hat{g}_{\lambda}(x_i)}{1-\{\mathbf{S}_{\lambda}\}_{ii}}\right]^2$$

6.6 局部回归

算法：$X=x_0$ 处的局部回归

1. 从 n 个训练数据中选取 x_i 与 x_0 最近的 k 个点（可定义 $s=k/n$）；
2. 在这个邻域中，为每一个点赋予权重 $K_{i0}=K(x_i,x_0)$，使得与 x_0 最远的点的权重为 0，最近的取最大的权重。除了这 k 个最近的点外的所有点权重为 0；
3. 采用上面的权重对 y_i 和 x_i 拟合加权最小二乘回归，寻找 $\hat{\beta}_0,\hat{\beta}_1$，最小化下面的函数：
$$\sum_{i=1}^{n}K_{i0}(y_i-\beta_0-\beta_1x_i)^2$$
4. x_0 的拟合值由下式给出：$\hat{f}(x_0)=\hat{\beta}_0+\hat{\beta}_1x_0$。

注意到，上面的算法第三步中，权重 K_{i0} 随 x_0 而变，也就是说对另一个点处的局部回归，我们需要拟合一个新的模型，取新的一系列权重。局部回归有时称为基于记忆（memory - based）的过程，因为它与最近邻法类似，当我们每次计算一个预测时，都需要全部的训练数据。

作局部回归时，我们需要选择几个东西，比如怎么定义权重函数 K，拟合线性、常数抑或二次回归呢？最重要的一个选择是上面算法第一步中的 s，称为 span。span 相当于光滑样条的调节参数 λ，它控制非线性拟合的灵活性：s 越小，拟合越是局部化也就是越波动，反之，如果 s 很大，就相当于采用全部训练数据的全局拟合了。

局部回归的思想可以推广种多种形式。比如，对于 p 个特性 $X_1,X_2,\cdots,X_p$ 的情形，我们可以作如下推广：对某些变量作全局回归，而对另一个变量（比如时间）作局部回归。这种模型称为“可变系数模型”（varying coefficient model），可用来对最近得到的数据施以修正。

6.7 广义加法模型

前面的扩展属于单变量模型，下面的广义加法模型（generalized additive model，GAM）是针对多元线性回归的扩展，可应用于定量和定性的应变量，亦即回归和分类问题。对于回归问题，我们将线性项 $\beta_j x_{ij}$ 替换为（光滑）线性函数 $f_j(x_{ij})$：

$$y_i = \beta_0 + f_1(x_{i1}) + f_2(x_{i2}) + \cdots + f_p(x_{ip}) + \epsilon_i$$

GAM 有以下优点：

- 自动拟合非线性关系，而无须尝试不同的变换；
- 非线性拟合可以提高预测准确性；
- 由于模型是可加的，具有比较好的解释性，适合推断；
- f_j 的光滑性可由自由度来概括。

GAM 的主要缺点是：只能表示加项，没有包括变量的重要交互项。然而，我们可以手工加入形如 $X_j \times X_k$ 的交互项。而且，我们还可加入形如 $f_{jk}(X_j, X_k)$ 的低维交互函数，这些项可以采用局部回归、二维样条等方式拟合。

7 决策树模型

7.1 回归树

回归树的构造方法：

1. 将特征空间分割为 J 个不同的互不重叠的区域 $R_1, R_2, \cdots, R_J$；
2. 对于落入 R_j 的所有观测，令其预测值都等于这些观测的响应值的均值。

如何分割区域？理论上可以分割成任意形状，但实际中为了易于处理和解释，一般将特征空间分割成高维矩形，称为箱子（boxes）。目标是找到 $R_1, \cdots, R_J$，最小化下面的 RSS：

$$\sum_{j=1}^{J} \sum_{i \in R_j} \left(y_i - \hat{y}_{R_j}\right)^2,$$

其中，$\hat{y}_{R_j}$ 为第 j 个箱子中所有响应的均值。考虑所有可能的划分是不现实的。我们采用一种称为“递归二项划分”（recursive binary splitting）的方法。该方法具有两个特征：（1）自上而下的（top-down）；（2）贪心的（greedy）：每一步都是当前的最优划分，不考虑将来的情况。递归二项划分可能导致过拟合，因为生成的树可能过于复杂。减少划分的次数或许能减少方差、增加解释能力（只需付出一点点偏差的代价）。一种可能的改进是，仅当 RSS 的减少超过某个（较高的）临界值时才作划分。这样能够生成较小的树，但太过于短视，因为或许一次看似无用的划分后面紧接着有一次非常好的划分（也就是说，有的划分会带来将来的较大的 RSS 改善）。因此，更好的策略是，先种一棵很大的数 T_0，再砍成子树（subtree)。怎么砍最好呢? 直观上看，我们的目标是选择能得到最低测试误差率的子树。给定一颗子树，可以利用交叉验证或验证集方法估计其测试误差。对所有可能的子树进行验证太耗时，所以我们需要选取一部分子树来考察。一种办法是成本复杂性修剪法（cost complexity pruning），也称为最弱连接修剪法（weakest link pruning)。考虑由一个非负调节参数（α）标记的树序列，每个 α 都对应一棵子树 $T \subset T_0$，使得下式尽可能小:

$$\sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T|$$

其中，$|T|$ 表示树 T 的终结点数目，R_m 为与第 m 个终结点对应的矩形，$\hat{y}_{R_m}$ 为与 R_m 对应的预测值。α 控制着子树复杂性与样本拟合能力之间的权衡。当 $\alpha = 0$，子树就等于 T_0；当 α 增加时，要使上式变小，只能选择较小的子树（终结点数目较小)。这种机制类似于 lasso。α 的选取可通过交叉验证或验证集方法来完成，然后回到完整数据集获得与 α 对应的子树。

算法：回归树的构造

1. 采用递归二项划分种一棵大树，仅当每个终结点的观测数少于某个最小值时停止划分；
2. 对大树应用成本复杂性修剪法，得到一个最优子树序列（α 的函数)；
3. 采用 K-折交叉验证选择 α。对于每个 $k = 1, \cdots, K$:

(a) 在 $(K-1)/K$ 比例的训练数据上重复步骤 1 和 2，不包括第 k 折；
(b) 在第 k 折的数据上计算预测均方误（α 的函数)。对上面的结果作平均，选择使平均值最小的 α。

4. 回到步骤 2 中与选取的 α 相对应的那个子树。

7.2 分类树

分类树的构造与回归树类似。但是，RSS 不能作为划分的标准。一种途径是用分类错误率代替 RSS。分类错误率定义为：

$$E = 1 - \max_k(\hat{p}_{mk}).$$

其中，$\hat{p}_{mk}$ 表示第 m 个区域中训练数据来自类别 k 的占比。但是，分类错误对树增长不够敏感，实践中常用另外两个指标。一种是 Gini 指数，定义为：

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}),$$

G 衡量 K 个类别的总方差。可以证明，当所有的占比 $\hat{p}_{mk}$ 接近 0 或 1 时，Gini 指数值比较小。另一个类似的指标是交叉熵（cross-entropy）：

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}.$$

实际上，G 和 D 的值非常接近。当构建分类树时，一般采用两种方式之一来评估划分的质量，因为它们对结点纯粹性的敏感度比分类错误率要高。这三种方法都可以用来修剪树，但如果追求最终树的预测准确性，应优先选用分类错误率。

7.3 树和线性模型的比较

线性模型与树模型风格迥异，前者假定模型形式为：

$$f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j,$$

后者假定模型形式为：

$$f(X) = \sum_{m=1}^{M} c_m \cdot 1_{(X \in R_m)}.$$

哪一种模型更好？取决于手里的问题。如果特征与响应之间近似为线性关系，选用线性模型比较好。如果特征与响应之间存在高度非线性或复杂关系时，决策树模型往往比传统方法效果好。

一般而言，树模型相比传统方法，有如下优点：

- 树非常容易解释，甚至比线性回归还容易解释；
- 决策树一定程度上更符合人类决策特点；
- 树可以通过图形来展示，易于被非专家看懂；
- 对于定量变量，树不需要构造哑变量即可处理。

不幸的是，树模型的预测准确性一般比传统方法要低。但是，利用随机森林、boosting 等方法把很多决策树加总，预测准确性可以大大提高。

7.4 Bagging

Bagging（Bootstrap aggregation）以 bootstrap 的思想为基础，通过对重复抽样多次拟合取平均来降低方差。首先，bootstrap 生成 B 个不同的训练集，从每一个训练集都得到一个预测 $\hat{f}^{*b}(x)$，接着对其平均得到：

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x).$$

Bagging 不仅能改善回归模型，对决策树尤其有用。我们可以利用 B 个自举训练集，生成 B 个回归树（不断生长，不修剪，从而这些树具有高方差、低偏差），对这些树取平均来降低方差。上面讲的是回归情况。对于分类问题，bagging 具有类似的功能，最简单的方法是，对给定的观测，记录 B 棵树各自的预测类别，然后用多数投票法来决定归属。树的书目 B 对于 bagging 来说并不是一个关键参数。实际运用中，一般取一个足够大的 B，使得测试误差开始降下来。

可以证明，平均来说，每一个树用到约 2/3 的观测。剩余的 1/3 观测称为 out-of-bag（OOB）观测。我们可以利用这些 OOB 作预测，对于第 i 个观测，将得到 $B/3$ 个预测，再对其作平均或应用多数投票，从而得到单个预测值。对所有 n 个观测都应用这个过程，可以计算出总体的 OOB MSE 或分类错误率。OOB 误差可以作为测试误差的估计，因为这些预测值是基于拟合外数据得到的。实际例子表明，当 B 足够大时，OOB 误差基本上等于 LOOCV 误差。OOB 方法对于交叉验证难以胜任的大型数据集特别有用。

可以发现，bagging 在改善预测准确性的同时，牺牲了模型的解释性。然而，我们可以通过 RSS（回归树）或 Gini 指数（分类树），获得变量重要性（variable importance）的整体概要。

7.5 随机森林

随机森林（random forests）通过对树进行去相关（decorrelate），改善自举集成树。与 bagging 类似，我们要利用自举训练集构造一系列决策树。不过，当考虑分割点时，我们从 p 个自变量中随机选取 $m \approx \sqrt{p}$ 个作为候选变量，然后仅使用其中一个。去相关的含义在于，bagging 倾向于每一次都选择重要的自变量，容易导致生成的树相关性很高，对一系列相关数取平均并不能减少太多的方差。随机森林则不同，它的做法使得平均有 $(p-m)/p$ 次分割不考虑重要自变量，从而其他自变量有更多机会。对相关性较低的树作平均，能减少方差且更稳定。当 $m = p$ 时，随机森林就是通常的 bagging。

7.6 Boosting

与 bagging 不同，boosting 不包含自举抽样，而是基于对原数据集的修正来拟合模型。在 boosting 中，树是序列（sequentially）生长的：每棵数都依赖前面生成的树的信息。

算法：回归树的 boosting

1. 令 $\hat{f}(x) = 0$，对训练集中所有的 i，$r_i = y_i$；
2. 对 $b = 1, 2, \cdots, B$，重复下列步骤：

(a) 对训练数据 (X, r)，拟合一个带有 d 次分割的数 $\hat{f}^b$；
(b) 修正 $\hat{f}$，即加上一个新树的削减版本：

$$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x)$$

(c) 修正残差：

$$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i)$$

3. 输出 boosted model：

$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x).$$

可见，boosting 中的树不是刻意艰难地去拟合，而是慢慢学习的（learn slowly）。给定当前的树，我们对其残差进行拟合，然后将生成的树加到拟合后的函数中以修正残差。每一个这样的树可以非常小，终结点由算法的 d 控制。通过对残差拟合小树，我们渐渐地对 $\hat{f}$ 在其表现不好的区域进行改进。缩减参数 λ 进一步减速，使得更加不同形状的树对残差起作用。一般

来说，渐渐学习的模型往往有好的表现。与 bagging 不同，boosting 中树对已出现的树具有很强的依赖性。

boosting 包含三个调节参数：

1. 树的数目 B。当 B 太大时，boosting 可能发生过拟合（这与 bagging 不同），虽然发生得比较慢。采用交叉验证选择 B。
2. 缩减参数 λ。它是一个较小的正数，控制 boosting 的学习速率。常见的值有 0.1 和 0.001，根据问题而定。很小的 λ 需要有很大的 B 配套。
3. 每棵树的分割数 d。控制 boosted ensemble 的复杂性。通常 $d = 1$ 表现不错，此时每棵树都是一个 stump，包含一个分割。这种情形的 boosted ensemble 就是拟合一个加法模型，因为每一项仅包含一个变量。更一般的，d 表示交互深度（interaction depth）。

8 支持向量机

8.1 最大间隔分类器

p 维空间的超平面定义为：

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0$$

考虑一个 $n \times p$ 数据矩阵 $\mathbf{X}$：

$$x_1 = \begin{pmatrix} x_{11} \\ \vdots \\ x_{1p} \end{pmatrix}, \cdots, x_n = \begin{pmatrix} x_{11} \\ \vdots \\ x_{1p} \end{pmatrix}$$

所有的观测属于两个类别，即 $y_1, \cdots, y_n \in \{-1, 1\}$。某个测试数据为 $x^* = \left(x_1^*, \cdots, x_p^*\right)^T$。我们采用分离超平面的方法来学习一个分类器。假定存在一个超平面能完全分离这些训练数据。分离超平面具有下面的性质：

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) > 0, i = 1, \cdots, n.$$

如果分离超平面存在，分类器构造就非常自然了，只要看 $f(x^*) = \beta_0 + \beta_1 x_1^* + \cdots + \beta_p x_p^*$ 的符号即可。可以发现，这样的分离超平面可能有无数个，如何选择最优的一个呢？这就是最大间隔超平面（maximal margin hyperplane）。

求解下面的最优化问题：

$$\max_{\beta_0,\beta_1,\cdots,\beta_p} M$$

s.t.

$$\sum_{j=1}^{p}\beta_j^2 = 1,$$

$$y_i(\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\cdots+\beta_p x_{ip}) \geq M, \forall i = 1,\cdots,n.$$

最大间隔超平面仅依赖于支持向量，其他观测的移动对超平面并无影响，只要这样的移动不越过 margin 设置的边界。

8.2 支持向量分类器

基于分离超平面的分类方法对很多实际的数据集都不适用，比如对于有些数据集，根本不存在这样的超平面；如果超平面相应的 margin 很窄，就会产生过拟合。此时，我们希望考虑这样一种分类器：它基于超平面，但这个超平面并不一定完全分离两个类别。这种分类器对于个别观测 robustness 较高，同时能对大多数训练数据进行更好的分类。也就是说，对几个观测误分类是值得的，可以对其余观测作效果更好的分类。支持向量分类器（support vector classifier），也称为软间隔分类器（soft margin classifier），允许某些观测处于 margin 的错误一边，甚至是超平面的错误一边。

支持向量分类器是下面最优化问题的解：

$$\max_{\beta_0,\beta_1,\cdots,\beta_p,\epsilon_1,\cdots,\epsilon_n} M$$

s.t.

$$y_i(\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\cdots+\beta_p x_{ip}) \geq M(1-\epsilon_i),$$

$$\epsilon_i \geq 0, \sum_{i=1}^{n}\epsilon_i \leq C, \sum_{j=1}^{p}\beta_j^2 = 1,$$

其中，C 是一个非负调节参数。ϵ_i 是松弛变量（slack variables），它允许观测处于错误的位置。如果 $\epsilon_i = 0$，观测位置正确；如果 $\epsilon_i > 0$，观测处于 margin 错误边，我们称第 i 个观测偏离了 margin；如果 $\epsilon_i > 1$，观测处于超平面错误边。

考虑调节参数 C。它限定了所有偏离的数目和程度，相当于某种预算集。当 $C = 0$，必定有 $\epsilon_1 = \cdots = \epsilon_n = 0$，即不允许有偏离 margin 的观测，

此时问题就等同于最大间隔超平面。当 $C > 0$，偏离超平面的观测数不得超过 C。当 C 增大（减少）时，我们对偏离的容忍度提高（降低），margin 变宽（窄）。实践中，C 通过交叉验证来确定，反映了偏差-方差权衡。

上面的优化问题有个非常有趣的性质：只有处于 margin 或偏离 margin 的点才影响超平面，这些点就称为“支持向量”（support vectors）；其他的点可以变动，只要仍保持处于 margin 正确一边，就不影响支持向量分类器。调节参数 C 越大，支持向量越多。

8.3 支持向量机

实际中我们常遇到非线性边界的情况。我们可以通过引入高阶项，将特征空间扩大。比如，将 p 个自变量扩展到 $2p$ 个自变量：

$$X_1, X_1^2, X_2, X_2^2, \cdots, X_p, X_p^2.$$

最优化问题变为：

$$\max_{\beta_0,\beta_{11},\beta_{12},\cdots,\beta_{p1},\beta_{p2},\epsilon_1,\cdots,\epsilon_n} M$$

s.t.

$$y_i \left(\beta_0 + \sum_{j=1}^{p} \beta_{j1} x_{ij} + \sum_{j=1}^{p} \beta_{j2} x_{ij}^2 \right) \geq M(1 - \epsilon_i),$$

$$\epsilon_i \geq 0, \sum_{i=1}^{n} \epsilon_i \leq C, \sum_{j=1}^{p} \sum_{k=1}^{2} \beta_{jk}^2 = 1.$$

支持向量机（SVM）就是支持向量分类器基于核函数（kernels）的扩展。可以证明，支持向量分类器的计算（上述最优化问题的解）仅涉及到观测的Ⓕ积（inner products），并不涉及观测值本身。向量 a 和 b 的Ⓕ积定义为 $\langle a, b \rangle = \sum_{i=1}^{r} a_i b_i$，因此观测 x_i 和 $x_{i'}$ 的Ⓕ积为：

$$\langle x_i, x_{i'} \rangle = \sum_{j=1}^{p} x_{ij} x_{i'j}$$

可以证明：

- 线性支持向量分类器可以表示为：

$$f(x) = \beta_0 + \sum_{i=1}^{n} \alpha_i \langle x, x_i \rangle,$$

其中，n 个参数 $\alpha_i, i = 1, \cdots, n$ 对应于各个观测。

- 要估计参数 $\alpha_1, \cdots, \alpha_n$ 和 β_0，只要知道所有观测间的🄵积，共有 $\binom{n}{2}$ 个。

注意到，计算 $f(x)$ 就是找到新观测 x 与所有训练数据 x_i 的🄵积。然而，只有支持向量对应的系数才非 0，因此，求和范围可以大大缩小：

$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i \langle x, x_i \rangle,$$

其中，$\mathcal{S}$ 为所有支持向量的指标集。

现在假定每一次出现🄵积时，我们都用下面的一般形式代替：

$$K(x_i, x_{i'}),$$

其中，K 为某种形式的函数，称之为核（kernel）。核用来度量两个观测之间的相似性。如果核取上面的🄵积形式（线性核），结果就是支持向量分类器。核也可取为多项式，比如：

$$K(x_i, x_{i'}) = \left(1 + \sum_{j=1}^{p} x_{ij} x_{i'j}\right)^d, \quad d \in \mathbb{Z}^+.$$

称之为 d 阶多项式核。当支持向量分类器由多项式核构成，生成的分类器就是支持向量机。此时，（非线性）函数形式为：

$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i K(x, x_i).$$

当然，还有其他形式的扩展。比如，辐射核（radial kernel）：

$$K(x_i, x_{i'}) = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2).$$

辐射核具有很强的局部行为特点，只有附近的观测点才会影响某个测试观测的类别。

8.4 多类别支持向量机

最流行的两种拓展方法是：一对一（one-versus-one）和一对全（one-versus-all）。

（TODO：支持向量机与 logistic 回归的关系?）

9 非监督学习

非监督学习中，我们只有一系列观测 $X_1, X_2, \cdots, X_p$，没有响应变量。目的是回答类似下面的问题：有没有什么好的方法对数据进行可视化？能不能依据某种标准，把变量或观测划分成不同的（有意义的）组？非监督学习往往带有主观性，没有明确简单的目标（比如预测），通常作为探索性数据分析（exploratory data analysis）的一部分。非监督学习的结果一般很难评估好坏。非监督学习在很多领域的重要性正在提高，比如癌症研究、电子商务、搜索引擎等。

9.1 主成分分析

主成分分析（principal components analysis，PCA）包括计算主成分并用其分析理解数据的过程，常用于数据可视化。

假定我们希望对包括 p 个特征、n 个观测的数据集进行可视化。当 p 较大时，用散点图不现实，需要寻找更好的方法，特别是要找到原数据集的某种低维表示，能捕捉尽可能多的信息。PCA 认为，尽管每一个观测都是 p 维的，但并不是所有的维度都有用。PCA 试图找到一小部分变异性较高的维度（主成分），这些维度都是原有 p 个特征的线性组合。

第一主成分是 p 个特征的最大方差正则化线性组合：

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \cdots + \phi_{p1}X_p$$

其中，正则化是指 p 个系数的平方和为 1，这些系数称为主成分的负载（loadings），放在一起组成主成分负载向量 $\phi_1 = (\phi_{11}, \phi_{21}, \cdots, \phi_{p1})^T$。正则化的目的是保证方差不会无限大。

对于 $n \times p$ 数据集 $\mathbf{X}$，怎么计算第一主成分？因为我们只关心方差，所以假设每个变量均值为 0（即 $\mathbf{X}$ 所有列的均值为 0）。我们寻找样本特征的某个线性组合：

$$z_{i1} = \phi_{11}x_{i1} + \phi_{21}x_{i2} + \cdots + \phi_{p1}x_{ip}$$

使其方差最大（满足系数正则化条件）。换言之，第一主成分负载变量是下面最优化问题的解：

$$\max_{\phi_{11},\cdots,\phi_{p1}} \left\{ \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{j1}x_{ij}\right)^2 \right\}, s.t. \sum_{j=1}^{p}\phi_{j1}^2 = 1.$$

上面的目标函数可以写成 $\frac{1}{n}\sum_i z_{i1}^2$。由于 $\frac{1}{n}\sum_i x_{ij}=0$，因此 $z_{11},\cdots,z_{n1}$ 的平均值为 0，从而，上面的目标函数就等于 z_{i1} 的方差。这些 z_{i1} 称为第一主成分的得分（scores）。

第一主成分的几何解释是，负载向量定义了特征空间中数据变动最多的方向，得分 $z_{11},\cdots,z_{n1}$ 就是数据点 $x_1,\cdots,x_n$ 在这个方向上的投影。

第二主成分是与 Z_1 不相关的所有线性组合中方差最大的那个。形式为:

$$z_{i2}=\phi_{12}x_{i1}+\phi_{22}x_{i2}+\cdots+\phi_{p2}x_{ip},$$

可以发现，Z_2 与 Z_1 不相关的条件等同于方向 ϕ_2 与 ϕ_1 正交（垂直）。要计算 ϕ_2，只有在上面的最优化问题中将 ϕ_1 替换为 ϕ_2，加上正交条件即可。

一旦有了主成分，我们可以对它们作图，从而生成数据的低维视图。比如，将得分向量 Z_1 对 Z_2 作图，Z_1 对 Z_3 作图，Z_2 对 Z_3 作图，如此等等。几何上看，就相当于将原始数据投射到由 ϕ_1,ϕ_2,ϕ_3 张成的子空间。

9.2 聚类方法