超饱和设计中的变量筛选

彭小令

(香港浸会大学数学系　　香港)

摘要：在工业生产和科学研究的很多领域中经常存在着因子个数多于试验次数的问题，而目前能够用在这种数据上的有效的变量筛选方法还很少。本文主要介绍了两种在现代多元统计中广泛应用的变量选择方法：LASSO 和 SCAD，以及如何将它们应用到超饱和试验据中筛选出比较重要的因子。

关键词：LASSO，SCAD，惩罚的最小二乘，超饱和设计。

1．介绍

在科学研究的许多领域，例如社会科学，生物学，化工学等，对于做回归分析可使用的理论模型较少。即使有理论模型，也可能包含不能直接量测的自变量（因子）。在这种情况下，研究人员不得不指望于那些可以得到的可能与应变量有关系的自变量。显然，这样的自变量是很多的，所以，在收集了很多的自变量的时候，我们就自然会面临着一个自变量选择的问题，因为其中的有些自变量可能对问题的研究并不重要，也可能实际上与其它自变量重迭。这时，我们所面临的问题就是怎样在这众多的自变量当中选出一组比较重要的来，这组自变量要足够少，以便得出的回归模型易于解释并且有较好的预测能力，它又必须充分多，以便对应变量能够进行合适的描述。

另一方面，在做具体的研究试验特别是工业试验和医学试验时，由于经费，试验条件等方面的原因，实际上做试验的次数受到很大程度的限制，而在一次实验中能够的到的潜在的影响因素却可能很多，甚至远远大与试验的次数。在这样的情况下，采用超饱和设计是一种常用的，并且行之有效的办法。现今超饱和设计正在受到越来越多的关注，例如 Lin（1995）介绍了一种产生系统的超饱和设计的方法。另外，Fang et al（2000）提出了一种通过准蒙特卡洛法创建多水平超饱和设计的方法。对超饱和设计中的数据进行分析，变量选择往往是第一步，也是重要的一步。而如何从大量的试验因子中筛选出少量对响应变量有影响的，重要的，又足够解释模型的因子，将是我们这篇文章将要讨论的内容。

在传统的多元统计分析中，有两种常用的变量筛选的方法：一种是最优子集法，即考虑所有可能的回归模型（由自变量的所有子集组成），再根据研究人员指定的标准，最终选出一个“最优”子集。这种方法看起来似乎很理想，但是也有它比较严重的缺陷，那就是计算量太大，由于需要搜索所有的子集，对于 p 个自变量，全部可能的子集数目就有 2^p-1 个之多，一般来说，当 p 的个数大于 30 就没有办法进行计算了。另一种则是逐步回归法，开始它将贡献最大的一个变量选入回归方程，并且预先确定两个阈值 F_{in} 和 F_{out}，用于决定变量能否入选或剔除。逐步回归在每一步有三种可能的功能：a) 将一个新变量引进回归模型，

这时相应的 F 统计量必须大于 F_{in}，b) 将一个变量从回归模型中剔除，这时相应的 F 统计量必须小于 F_{out}，c) 将回归模型内的一个变量和回归模型外的一个变量交换位置。但是这种逐步筛选的方法也有它不足的地方，那就是它的不稳定性 [Breiman, 1996]，另外它对于超饱和设计也不大适宜 [Westfall et al, 1998]。

在这篇文章中我们将介绍两种比较新的，基于惩罚的最小二乘的变量选择方法: (1) .“ 最小的绝对缩减和变量选择算子 ”(least absolute shrinkage and selection operator)，简称 LASSO [Tibshirani, 1996] ；(2) .“ 绝对偏差的平滑缩减 ”(smoothly clipped absolute deviation)，简称 SCAD [Fan 和 li, 2001]。这两种方法不仅在现代研究的各个领域中被广泛应用，经过一定的改进以后还可以用于超饱和设计的变量筛选，并取得了一些好的效果。

2． 惩罚的最小二乘

2．1 几种惩罚最小二乘的定义

首先我们考虑最一般的线形回归模型：

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{2.1}$$

其中 $\mathbf{y}$ 是一个 $n\times 1$ 的响应变量，$\mathbf{X}$ 是一个 $n\times p$ 的设计矩阵，而 $\boldsymbol{\varepsilon}$ 表示残差。标准最小二乘对系数 $\boldsymbol{\beta}$ 的估计是通过最小化残差平方和得到。但是，当变量的个数较多时，这种经典的方法在预报的精确性方面不太令人满意。这是由于标准最小二乘经常会因为追求解的无偏性而使得其估计具有较大的方差，从而导致较大的预报误差。而预报的精确性有时候可以通过对系数进行缩减，使某些比较小的系数自动设为零来得到提高。我们通过牺牲了一些偏差来减少预报值的方差，因此可以从总体上提高模型的预报能力。实际中的系数缩减一般则是通过对最小二乘的系数进行惩罚（即惩罚的最小二乘）来实现。为了陈述的简便，不失一般性地我们假设设计矩阵 $\mathbf{X}$ 是正交的。令 $\mathbf{z} = \mathbf{X}^{\mathrm{T}}\mathbf{y}$， $\hat{\mathbf{y}} = \mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{y}$，惩罚的最小二乘可写成如下形式：

$$\begin{aligned} &\tfrac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\sum_{j=1}^{d} p_j(|\beta_j|) \\ &= \tfrac{1}{2}\|\mathbf{y}-\hat{\mathbf{y}}\|^2 + \tfrac{1}{2}\sum_{j=1}^{d}(z_j-\beta_j)^2 + \lambda\sum_{j=1}^{d} p_j(|\beta_j|) \end{aligned} \tag{2.2}$$

为简便起见，我们假设实施在所有的系数上的惩罚函数 $p_j(\cdot)$ 是相同的，记为 $p(|\cdot|)$，并记 $\lambda p(|\cdot|)$ 为 $p_\lambda(|\cdot|)$，以表示惩罚函数跟 λ 有关。这样，使（2.2）式达到最小实际上就是求 θ，使得

$$\tfrac{1}{2}(z-\theta)^2 + p_\lambda(|\theta|) \tag{2.3}$$

达到最小。不同的惩罚函数，得到的解的形式也不同。对于硬门限（hard

thresholding）惩罚函数

$$p_\lambda(|\theta|) = \lambda^2 - (|\theta| - \lambda)^2 I(|\theta| < \lambda) \tag{2.4}$$

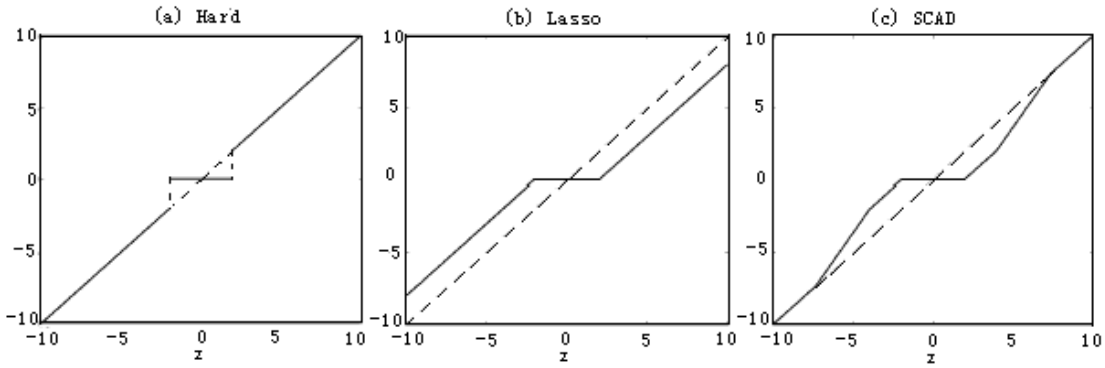
[Fan, 1997] 给出解的形式为：

$$\hat{\theta} = zI(|z| > \lambda) \tag{2.5}$$

参见图一 (a)。而由 L_1 惩罚 $p_\lambda(|\theta|) = \lambda|\theta|$ 得到的则称作软门限（soft thresholding）法则:

$$\hat{\theta}_j = \text{sgn}(z_j)(z_j - \lambda)_+ \tag{2.6}$$

参见图一 (b)。这个结果由 Donoho 和 Johnstone（1994）得到，这与 LASSO 的结果恰好是相同的 [Tibshirani, 1996]。它通过对较大的系数进行压缩并使另外的系数缩减为零，从而达到变量选择的目的并期望达到较好的预报效果。



图一：三个门限函数的比较（a）硬门限（b）L_1（c）SCAD（$\lambda = 2$，$a = 3.7$）

什么样的惩罚函数才算是好的惩罚函数呢？在 Fan 和 Li（1997）的文章中 指出：对于一个好的惩罚函数，它得出的解应该具有下面的三个性质：

（1）无偏性：当未知参数的真实值比较大时，所得的解应该近似地无偏，这样可以避免不必要的模型偏差。

（2）稀疏性：由于我们想得到的是一个门限法则，所以这个解应该能够自动地将小的估计系数设为零，来降低模型的复杂程度。

（3）连续性：所得的解应该保证其与最小二乘解 z 之间的连续性，以避免模型在预报时的不稳定性。

然而，前面提到的两种惩罚函数都不同时具有这三条性质。为此，Fan 和 Li 将连续可微的惩罚函数

$$p_\lambda'(\theta) = \lambda\{I(\theta \le \lambda) + \frac{(a\lambda - \theta)_+}{(a-1)\lambda} I(\theta > \lambda)\}，\text{其中} a > 2，\ \theta > 0 \tag{2.7}$$

应用到惩罚的最小二乘，并希望能改进前面提到的硬门限惩罚和 L_1 惩罚所得到的解的性质，由（2.7）式定义的惩罚函数被简称为 SCAD。它的解由 Fan (1997) 给出：

$$\hat{\theta}=\begin{cases}\operatorname{sgn}(z)(|z|-\lambda)_+ & |z|\le 2\lambda \\ \{(a-1)z-\operatorname{sgn}(z)a\lambda\}/(a-2) & 2\lambda<|z|\le a\lambda \\ z & |z|>a\lambda\end{cases} \tag{2.8}$$

参见图一 (c)。由图一的比较我们可以看出，硬门限法则不具有连续性的性质而 L_1 门限函数则不能保证在参数值较大时的无偏性，只有 SCAD 能同时具备这三种性质。这种方法在选择变量的同时对系数进行估计，且求得的估计值等效于已知正确的子模型。通过大量的模拟实验，也证实了 SCAD 惩罚门限优于其它的变量选择方法。

2．2　LASSO 和 SCAD 的算法

对于 LASSO，Tibshirani 在 1996 年提出了一种求解的算法 [Tibshirani，1996]，之后，
Fu（1998）又为 LASSO 提供了一种“ shooting ”算法（http://lib.stat.cmu.edu/S/），这种算法比较于之前的收敛速度快了很多。 该算法如下：

（1） 取迭代初始值 $\hat{\boldsymbol{\beta}}_0=\hat{\boldsymbol{\beta}}_{OLS}=(\hat{\beta}_1,\cdots,\hat{\beta}_p)^T$。

（2） 在第 m 步，对所有的 $j=1,\cdots,p$，令 $S_0=S_j(0,\hat{\beta}^{-j},X,y)$，则

$$\hat{\beta}_j=\begin{cases}\dfrac{\lambda-S_0}{2x_j{}^Tx_j} & S_0>\lambda \\ \dfrac{-\lambda-S_0}{2x_j{}^Tx_j} & S_0<\lambda \\ 0 & |S_0|\le\lambda,\end{cases}$$

经过 p 次计算，用新的估计值 $\hat{\beta}_m=(\hat{\beta}_1,\cdots,\hat{\beta}_p)^T$ 替换了上一步的结果。

（3） 重复第二步直到 $\hat{\beta}_m$ 收敛。

利用 Newton-Raphson 方法，Fan 和 Li 为 SCAD 提供了一个快速的迭代算法，并证明了该算法在初始值靠近真实的解的时候能够迅速地收敛。该算法如下：

（1） 取迭代初始值 $\hat{\boldsymbol{\beta}}_0=\hat{\boldsymbol{\beta}}_{OLS}=(\hat{\beta}_1,\cdots,\hat{\beta}_p)^T$。

（2） 新的估计值 $\boldsymbol{\beta}_1=\{\mathbf{X}^T\mathbf{X}+n\sum_\lambda(\boldsymbol{\beta}_0)\}^{-1}\mathbf{X}^Ty$ ，

其中 $\sum_\lambda(\boldsymbol{\beta}_0)=\operatorname{diag}\{p'_\lambda(|\beta_{10}|)/|\beta_{10}|,\cdots,p'_\lambda(|\beta_{d0}|)/|\beta_{d0}|\}$。

（3） 重复第二步直到 $\boldsymbol{\beta}_1$ 收敛。

2．3　参数的选取

一般来说，这类问题的参数选取都是采取交互验证的方式，例如 5-fold 或

10-fold 的交互验证，也就是说，把训练样本随机分成 5 份或 10 份，每次选取一份用来做预测，剩下的用来做回归得到系数的估计值，最后选取一个使得平均预报效果最好的参数。大量的实践证明：这种由数据本身驱策而得到的参数往往能得到比较好的结果。对于 SCAD 惩罚函数中的参数 a，从贝叶斯风险分析中可以看出，贝叶斯风险对于 a 的取值并不敏感，而在变量个数小于 100 的时候，$a = 3.7$ 是一个比较理想的选择。

3． 改进的 LASSO 和 SCAD

从上面的两个算法可以看出，它们都是选取最小二乘的解作为初始迭代值。这就使得这两种方法都不能直接用于超饱和设计的数据，即设计矩阵不满秩。针对这种情况，öjelund et al（2001）提出了改进的 LASSO 算法，该算法同样采用了 Newton-Raphson 方法，其主要思想是将拉格朗日项 $\lambda \sum_j |\beta_j|$ 做了一个迭代的二次近似以确保可微。同时，在初始值的选取问题上，öjelund，et al 采用了 $\boldsymbol{\beta}_0 = (\mathbf{X}^T\mathbf{X})^- \mathbf{X}^T\mathbf{y}$，这里 $(\mathbf{X}^T\mathbf{X})^-$ 表示的是 Moore-Penrose 广义逆。而对于 SCAD，Li 和 Lin（2002）在“超饱和设计数据分析”一文中讨论并证明了它在超饱和设计的数据中具有同样优良的性质和收敛速度。由于初始值要与真实值比较靠近，他们建议采用一些逐步的变量选择的方法来求得一个初始值，并且值得注意的是，这些方法最好选取比较宽松的门限以便能够将重要的变量都包括进来。改进后的这两种方法都能够直接地运用到设计矩阵不满秩的情况并选出其中重要的变量。

4．总结

在现代多元统计领域，针对满秩的设计矩阵的回归和变量选择方法有很多，研究得也比较深入，但是专门针对超饱和设计和别的不满秩的设计矩阵的变量选择方法还比较少。而随着科学技术的发展，特别是医学和生物学研究的不断进步，研究者遇到这种变量个数大于试验次数的情况也越来越多，有时候甚至做一次试验能测得上千个参数值，而受经济等方面的影响，试验却往往只能重复几次。怎样从有限的响应值判断出哪些参数是真正重要的就成为研究者们面临的一大难题。在这篇文章中介绍的两种变量选择的方法 LASSO 和 SACD 都是在对传统的变量选择方法进行改进后得到的，除了吸取传统变量选择方法的一些优点，还具有一些好的性质，故而被广泛地应用到了各门应用学科当中。并且，这两种方法在经过改进后都能够被用于设计矩阵不满秩的数据并在模拟数据和真实数据中都取得了比较理想的效果。由于本文完全是从介绍的角度给出了这两种方法的产生，定义和算法，目的是让想使用这两种方法的研究者们对它们有一个初步的认识，所以本文并没有列出一些应用的结果，有关的这些和细节可以到相关文献上查找。

参考文献：

[1] Breiman, L., 1996. Heuristics of Instability and Stabilization in Model Selection. The Annals of Statistics, 24, 2350–2383.

[2] Donoho, D. L., and Johnstone, I. M.，1994. Ideal Spatial Adaptation by Wavelet shrinkage. Biometrika 81, 425–455.

[3] Fan, J., 1997. Comments on “Wavelets in statistics: a review” by A. Antoniadis. J. Italian Statist. Assoc. 6, 131－138.

[4] Fan, J., Li, R., 2001. Variable selection via nonconcave penalized likelihood and its oracle properties. J. Amer. Statist. Assos. 96, 1348－1360.

[5] Fang, K.T., Lin, D.K.J., Ma, C.X., 2000. On the construction of multi-level supersaturated designs. J. Statist. Plan. Inference 86, 239－252.

[6] Fu, W.J., 1998. Penalized regression: the bridge versus the LASSO. J. Comput. Graphical Statist. 7, 397－416.

[7] Li, R.Z., Lin, D.K.J., 2002. Data analysis in supersaturated designs. Statistics & Probability Letters 59, 135-144.

[8] Lin, D.K.J., 1995. Generating system supersaturated designs. Technometrics 37, 213－225.

[9] öjelund, H., Madsen, H., Thyregod, P., 2001. Calibration with absolute shrinkage. Journal of Chemometrics 15, 497-509.

[10] Tibshirani, R., 1996. Regression shrinkage and selection via the LASSO. Journal of the Royal Statistical Society, B, 58, 267-288.

[11] Westfall, P.H., Young, S.S., Lin, D.K.J., 1998. Forward selection error control in analysis of supersaturated designs. Statist. Sinica 8, 101－117.