Sparse inverse covariance estimation with the graphical lasso

摘要

我们考虑通过对逆协方差矩阵应用套索惩罚来估计稀疏图的问题。使用套索的坐标下降过程，我们开发了一个简单的算法——图套索（graphical lasso），该算法非常快速：它在不到一分钟的时间内解决了一个1000个节点的问题（约500,000个参数），比竞争方法快30-4000倍。它还提供了精确问题与Meinshausen和Bühlmann（2006）所建议的近似之间的概念联系。我们在一些来自蛋白质组学的细胞信号数据上说明了该方法的效果。

1 Introduction

近年来，许多作者提出使用 L_1（套索）正则化来估计稀疏无向图模型。连续数据的基本模型假设观测值具有均值 μ 和协方差矩阵 Σ 的多元高斯分布。如果 Σ 的第 i 行第 j 列元素为零，则变量 i 和 j 在给定其他变量的条件下是独立的。因此，对于估计 Σ 的逆矩阵的非零元素增加稀疏性，施加 L_1 惩罚是合理的。

Meinshausen和Bühlmann（2006）对这个问题采取了简单的方法；他们通过将lasso模型拟合到每个变量上，使用其他变量作为预测变量来估计一个稀疏图模型。然后，如果变量 i 对 j 的估计系数或变量 j 对 i 的估计系数不为零（或者他们采用了AND规则），则估计的 Σ_{ij}^{-1} 被估计为非零。他们证明，在渐近情况下，这一方法一致地估计了 Σ^{-1} 的非零元素集合。

其他作者提出了用于精确最大化L1惩罚对数似然的算法；Yuan和Lin（2007）、Banerjee等人（2007）以及Dahl等人（2007）采用内点优化方法来解决这个问题。两篇论文还证明了Meinshausen和Bühlmann（2006）的简单方法可以看作是精确问题的一种近似。 我们使用Banerjee等人（2007）的分块坐标下降方法作为起点，并提出了一个新的算法来解决精确问题。这种新的过程非常简单，在我们的测试中比竞争方法快得多。它还弥合了（Meinshausen和Bühlmann，2006）的建议与精确问题之间的“概念鸿沟”。

2 提议的方法

假设我们有 N 个维度为 p 的多元正态观测值，其均值为 μ，协方差矩阵为 Σ。按照Banerjee等人（2007）的方法，令 $\Theta = \Sigma^{-1}$，并且令 S 为经验协方差矩阵，问题是在非负定矩阵 Θ 上最大化受惩罚的对数似然

$$\arg\max_{\Theta \in \mathbb{R}^{N \times N}} (\log \det \Theta - \operatorname{tr}(S\Theta) - \rho \|\Theta\|_1) \tag{1}$$

在这里，tr表示迹（trace），$\|\Theta\|_1$ 表示 L_1 范数，即 Σ^{-1} 的元素的绝对值之和。式（1）是数据的高斯对数似然函数，针对均值参数 μ 进行了部分最大化。Yuan和Lin（2007）使用Vandenberghe等人（1998）提出的“maxdet”问题的内点方法来解决此问题。Banerjee等人（2007）开发了一种不同的优化框架，为我们的工作提供了动力。

Banerjee等人（2007）表明问题（2.1）是凸的，并考虑以下方式对 Σ（而不是 Σ^{-1}）进行估计。设 W 为 Σ 的估计值。他们证明可以通过以分块坐标下降的方式在 W 的每一行和相应列上进行优化来解决这个问题。将 W 和 S 进行分块划分，

$$W = \begin{pmatrix} W_{11} & w_{12} \\ w_{12}^T & w_{22} \end{pmatrix}, S = \begin{pmatrix} S_{11} & s_{12} \\ s_{12}^T & s_{22} \end{pmatrix} \tag{2}$$

他们证明 w_{12} 的解满足

$$w_{12} = \arg\min_y \{ y^T W_{11}^{-1} y : \|y - s_{12}\|_\infty \le \rho \} \tag{3}$$

这是一个有约束的二次规划问题（QP），他们使用内点过程来解决它。通过对行和列进行排列，使目标列始终为最后一列，他们为每一列解决类似于（2.3）的问题，并在每个阶段更新 W 的估计值。直到收敛为止，重复此过程。如果该过程初始化为正定矩阵，他们表明即使 $p > N$，该过程的迭代仍保持正定且可逆。

利用凸对偶性，Banerjee等人（2007）继续展示了解决（3）与解决下面的对偶问题等价：

$$\min_{\beta}\{\frac{1}{2}\|W_{11}^{1/2}\beta - b\|^2 + \rho\|\beta\|_1\} \tag{4}$$

其中 $b = W_{11}^{1/2} s_{12}$；如果 β 是 (4) 的解，那么 $w_{12} = W_{11}\beta$ 是 (3) 的解。表达式（4）类似于套索回归，并且是我们方法的基础。

首先，我们直接验证了解（1）和（4）之间的等价性。展开关系式 $W\Theta = I$ 会给出一个下面会用到的表达式：

$$\begin{pmatrix} W_{11} & w_{12} \\ w_{12}^T & w_{22} \end{pmatrix} \begin{pmatrix} \Theta_{11} & \theta_{12} \\ \theta_{12}^T & \theta_{22} \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0^T & 1 \end{pmatrix}. \tag{5}$$

现在，最大化对数似然函数（1）的次梯度方程是：

$$W - S - \rho \cdot \Gamma = 0 \tag{6}$$

利用对数行列式 $\log\det\Theta$ 的导数等于 $\Theta^{-1} = W$ 的事实（例如，参考Boyd和Vandenberghe（2004,p641）），我们可以得到上面的次梯度方程。其中 $\Gamma_{ij} \in \mathrm{sign}(\Theta_{ij})$；即如果 $\Theta_{ij} \neq 0$，则 $\Gamma_{ij} = \mathrm{sign}(\Theta ij)$，否则 $\Gamma_{ij} \in [-1, 1]$。

现在，（6）的右上角块为：

$$w_{12} - s_{12} - \rho \cdot \gamma_{12} = 0 \tag{7}$$

另一方面，根据（2.4），次梯度方程可以推导为：

$$W_{11}\beta - s_{12} + \rho \cdot \nu \tag{8}$$

其中，$\nu \in \mathrm{sign}(\beta)$ 是逐元素地取符号函数。假设 (W, Γ) 是 (6) 的解，即 (w_{12}, γ_{12})是 (7) 的解。那么有 $\beta = W_{11}^{-1} w_{12}$ 和 $\nu = -\gamma_{12}$ 是 (8) 的解。第一个和第二个项的等价性是显然的。对于符号项，由于根据（5）有 $W_{11}\theta_{12} + w_{12}\theta_{22} = 0$，我们得到 $\theta_{12} = -\theta_{22} W_{11}^{-1} w_{12}$。由于 $\theta_{22} > 0$，因此有 $\mathrm{sign}(\theta_{12}) = -\mathrm{sign}(W_{11}^{-1} w_{12}) = -\mathrm{sign}(\beta)$。这证明了等价性。我们注意到，lasso问题（4）的解 β 给出了（除了一个负常数外） Θ 的相应部分：$\theta_{12} = -\theta_{22}\beta$。

现在来看本文的主要观点。问题（2.4）看起来像是一个套索（L1正则化）最小二乘问题。实际上，如果 $W_{11} = S_{11}$，那么解 $\hat{\beta}$ 很容易看出等于其他变量上的第 p 个变量的套索估计，并与Meinshausen和Bühlmann（2006）的提议相关。如Banerjee等人（2007）所指出的，通常情况下 $W_{11} \neq S_{11}$，因此Meinshausen和Bühlmann（2006）的方法无法得到极大似然估计。他们指出他们的分块内点过程等价于递归地解决和更新套索问题（4），但没有追求这种方法。我们利用这个方法有极大的优势，因为快速坐标下降算法（Friedman等人，2007）使得解决套索问题非常有吸引力。

从内积的角度来看，关于其他变量的第 p 个变量的通常套索估计使用的是数据 S_{11} 和 s_{12} 作为输入。为了解决（4），我们使用的是 W_{11} 和 s_{12} ，其中 W_{11} 是我们当前对 W 的上方块的估计。然后我们更新 w 并循环遍历所有的变量，直到收敛。

需要注意的是，根据（6），对于所有的 i，解 $w_{ii} = s_{ii} + \rho$，因为 $\theta_{ii} > 0$，所以 $\Gamma_{ii} = 1$。为了方便起见，我们将这个算法称为图形套索（graphical lasso）。以下是详细的算法步骤：

Graphical lasso algorithm

1. 以 $W = S + \rho I$ 为初始值。在接下来的步骤中，W 的对角线保持不变。

2. 对于每个 $j = 1, 2, \ldots, p$，$1, 2, \ldots, p$，$\ldots$，解决套索问题（4），该问题的输入是内积 W_{11} 和 s_{12} 。这将给出一个长度为 $p-1$ 的向量解 $\hat{\beta}$。使用 $w_{12} = W_{11}\hat{\beta}$ 来填充相应的行和列。

3. 继续进行直到收敛。

这个过程有一个简单而直观的视角。给定数据矩阵 $\mathbf{X}$ 和目标向量 $\mathbf{y}$，我们可以将线性最小二乘回归估计 $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$ 看作不是基于原始数据的函数，而是基于内积 $\mathbf{X}^T\mathbf{X}$ 和 $\mathbf{X}^T\mathbf{y}$ 的函数。类似地，也可以证明套索估计也是这些内积的函数。因此，在当前问题中，我们可以将对其他变量的第 p 个变量的套索估计看作具有如下函数形式：

$$\texttt{lasso}(S_{11}, s_{12}, \rho) \tag{9}$$

但是，对每个变量应用套索并不能解决问题（1）；为了通过图形套索来解决这个问题，我们使用内积 W_{11} 和 s_{12} 。也就是说，我们用下面的式子替换（9）：

$$\texttt{lasso}(W_{11}, s_{12}, \rho) \tag{10}$$

关键在于问题（1）不等同于 p 个单独的正则化回归问题，而是等同于 p 个耦合的套索问题，它们共享相同的 W 和 $\Theta = W^{-1}$。在适当的方式下，使用 W_{11} 代替 S_{11} 将信息共享在这些问题之间。 需要注意的是，在步骤（2）中的每次迭代都意味着对行和列进行排列，使目标列成为最后一列。上述第（2）步中的套索问题可以通过坐标下降法高效解决（Friedman等人，2007；Wu和Lange，2007）。以下是详细步骤：

令 $V = W_{11}$ 和 $u = s_{12}$，那么更新形式如下：

$$\hat{\beta}_j \leftarrow \frac{S(u_j - \sum_{k \neq j} V_{kj}\hat{\beta}_k\ ,\ \rho)}{V_{jj}}, \tag{11}$$

对于 $j = 1, 2, \ldots, p$，$1, 2, \ldots, p$，$\ldots$ 利用软阈值操作符 S 进行更新，其中S的定义如下：

$$S(x, t) = \text{sign}(x)(|x| - t)_+. \tag{12}$$

我们循环遍历预测变量直到收敛。在我们的实现中，当 W 的平均绝对变化小于 $t \cdot \text{ave}|S^{-\text{diag}}|$ 时，过程停止，其中 $S^{-\text{diag}}$ 是经验协方差矩阵 S 的非对角元素，t 是一个固定的阈值，默认设置为 0.001。

需要注意的是，$\hat{\beta}$ 通常是稀疏的，因此计算 $w_{12} = W_{11}\hat{\beta}$ 将会很快；如果有 r 个非零元素，则需要 rp 次操作。

虽然我们的算法估计了$\hat{\Sigma} = W$，但我们可以相对廉价地恢复$\hat{\Theta} = W^{-1}$ 。根据（5）中的分区，我们有以下关系：

$$\begin{aligned} W_{11}\theta_{12} + w_{12}\theta_{22} &= 0, \\ w_{12}^T\theta_{12} + w_{22}\theta_{22} &= 1, \end{aligned}$$

从中我们可以得到标准的分块逆表达式：

$$\theta_{12} = -W_{11}^{-1}w_{12}\theta_{22}, \tag{13}$$

$$\theta_{22} = 1/(w_{22} - w_{12}^T W_{11}^{-1} w_{12}). \tag{14}$$

由于 $\hat{\beta} = W_{11}^{-1}w_{12}$，我们有 $\hat{\theta}_{22} = 1/(w_{22} - w_{12}^T\hat{\beta})$ 和 $\hat{\theta}_{12} = -\hat{\beta}\hat{\theta}_{22}$。因此，$\hat{\theta}_{12}$ 是通过$-\hat{\theta}_{22}$ 简单缩放 $\hat{\beta}$ 得到的，计算起来很容易。虽然这些计算可以包含在图形套索算法的第2.2步中，但直到最后才需要它们；因此，我们将每个问题的所有系数 β 存储在一个 $p{\times}p$ 矩阵$\hat{B}$ 中，并在收敛后计算 $\hat{\Theta}$。

有趣的是，如果 $W = S$，那么这些只是获取分块矩阵的逆的公式。也就是说，如果我们将 $W = S$ 和 $\rho = 0$ 设置在上述算法中，那么一次遍历预测变量就可以计算出 S 的逆，每个阶段使用线性回归。

注释2.1 在某些情况下，为每个变量指定不同的正则化程度，甚至允许对每个逆协方差元素进行不同的惩罚，可能是有意义的。因此，我们最大化对数似然函数

$$\log\det\Theta - \mathrm{tr}(S\Theta) - \|\Theta * P\|_1 \tag{15}$$

其中 $P=\{\rho_{jk}\}$， $\rho_{jk}=\rho_{kj}$，并且 $*$ 表示分量逐元素相乘。很容易证明，通过前面的算法，在软阈值步骤（11）中将 ρ 替换为 ρ_{jk} ，可以最大化（15）。通常情况下，可以取 $\rho_{jk}=\sqrt{\rho_j\rho_k}$ 来为每个变量指定不同的正则化程度，其中$\rho_1,\rho_2,\dots,\rho_p$ 是一些给定的值。

注释2.2 如果在（1）的惩罚项中不考虑对角元素，则 w_{ii} 的解简单地为 s_{ii}，其他情况下算法与之前相同。