Binary Knockoffs Notes

Aaron Maurer

June 17, 2015

1 Preliminaries

Note: I will try to hold to the convention that X is the $n \times p$ data matrix, while $\mathbf{x}$ is the random vector variable from which each row of X was drawn. Accordingly, $\tilde{X}$ will be the knockoff matrix while $\tilde{\mathbf{x}}$ is a random vector variable. x_i will be the random variable corresponding to the ith entry of $\mathbf{x}$, while X_i is the vector of observations drawn from x_i in the data matrix.

Some early investigation into deterministic knockoffs (as described in the original knockoff paper) reveal that they don't perform well in L1 regularized logistic regression. Even when X_i is a null predictor of y, the X_i still tend to enter the model prior to $\tilde{X}_i$. The issue is that even when $x \sim N_p(\mathbf{0}, \Sigma)$ for some $\Sigma \succeq 0$, $\tilde{x}$ is not normally distributed. This can be seen from producing qq plots of X_i vs $\tilde{X}_i$ for each i. Of course, when X is a binary vector, $\tilde{X}$ completely doesn't match its distribution, causing the original X to beat the knockoffs into the model. This indicates that a new method of generating $\tilde{X}$ must be created to control FDR via knockoffs with regularized logistic regression.

2 Probabilistic Random Bernoulli Knockoffs

My idea is to generate $\tilde{\mathbf{x}}$ randomly such that, approximately, $\tilde{\mathbf{x}} \sim \mathbf{x}$. In particular, both variables should have similar marginal densities, expectations, and second moments. However, $\tilde{\mathbf{x}} \mid \mathbf{x}$ should also have desired knockoff property that $\mathrm{E}(\tilde{\mathbf{x}}'\mathbf{x} \mid \mathbf{x}) = \mathbf{x}'\mathbf{x} - \mathbf{s}$, where $\mathrm{diag}(\mathbf{x}'\mathbf{x}) - \mathbf{s}$ is small. In the general case, this is likely infeasible; however, if $\mathbf{x}$ is a binary vector, as is often the case, we know we are dealing with a much more limited class of random variables, and it should be possible to randomly generate $\tilde{\mathbf{x}} \mid \mathbf{x}$ so as to have the desired properties. At worst, this method will provide a suitable replacement for deterministic $\tilde{\mathbf{x}}$ for use with LASSO, and if we are lucky, it will work reasonably for other regularized GLMs.

3 Random Bernoulli Generation

Thankfully, there has been a reasonable amount of work on how one can generate random Bernoulli vectors with some kind of correlation among among the values. A random Bernoulli vector $\mathbf{x}$ can be summarized by its first two moments: a mean vector $\mathrm{E}(\mathbf{x}) = \mathbf{m} \in (\mathbf{0}, \mathbf{1})^{\mathbf{p}}$ and cross-moment matrix $\mathrm{E}(\mathbf{x}\mathbf{x}') = \mathbf{M} \in (\mathbf{0}, \mathbf{1})^{\mathbf{p}\times\mathbf{p}}$. Obviously, $m_i = \mathrm{P}(x_i = 1)$, $M_{ij} = \mathrm{P}(x_i = x_j = 1)$, and $m = \mathrm{diag}(M)$. For an arbitrary symmetric M to be valid cross-moment matrix, $M - mm' \succeq 0$, and

$$\max\{0, m_i + m_j - 1\} \leq M_{ij} \leq \min\{m_i, m_j\}$$

for all $i \neq j$[1]. Given a qualifying M, or observed X, there are a few ways of generating more random $\mathbf{x}$.

3.1 Gaussian Copula Family

Since multivariate normal distributions are easy to randomly draw, the idea is to find some random normal variable $z \sim N_p(\mathbf{0}, \Sigma)$ such that,for $x_i = I(z_i < 0)$, x has the desired properties. There are a number of ways to do this[2][3], but it turns out that there is only certain to exist a working Σ in the bivariate case.

3.2 μ-Conditionals family

It turns out that there exists a more flexible family which will always work for arbitrary M called μ-conditionals. The basic idea is that the X is generate sequentially as

$$\mathrm{P}(x_i = 1 \mid x_1, ..., x_{i-1}) = \mu\left(a_{ii} + \sum_{k=1}^{i-1} a_{ik} x_i\right)$$

for some monotone function $\mu : \mathbb{R} \to [0, 1]$. This is essentially a binomial family GLM for a link function μ. If one takes all of the a_{kj}, they can form a lower triangular matrix A, and then the joint density can be expressed as

$$\mathrm{P}(\mathbf{x} = \gamma) \propto \mu(\gamma' A \gamma)$$

If μ is chose such that it is a bijection and differentiable, there is a unique M such that $\mathrm{E}(x_i x_i') = M$[4]. It turns out that the natural choice for μ is the logistic link function, which yields the Ising model, the "binary analogue of the multivariate normal distribution which is the maximum entropy distribution on $\mathbb{R}^p$ having a given covariance matrix." Additionally, it has the usual benefit that the coefficients can be viewed as a log odds ratio:

$$a_{ij} = \log\left(\frac{\mathrm{P}(x_j = x_k = 1)\mathrm{P}(x_j = x_k = 0)}{\mathrm{P}(x_j = 0, x_k = 1)\mathrm{P}(x_j = 1, x_k = 0)}\right)$$

when $i \neq j$. I think this dictates that if $\mathbf{x}$ is generated from this model with $a_{ij} = 0$, then x_i and x_j are independent.

There is no closed form to calculate the entries in A if $p > 1$, but they can be derived numerically two ways.

[1] "On parametric families for sampling binary data with specified mean and correlation" - http://arxiv.org/abs/1111.0576

[2] "On the Generation of Correlated Artificial Binary Data" - http://epub.wu.ac.at/286/1/document.pdf

[3] "On parametric families for sampling binary data with specified mean and correlation"

[4] "On parametric families for sampling binary data with specified mean and correlation"

1. If one is attempting to replicate the empirical cross-moments from a data matrix X, a_{1i} to a_{ii} can be derived from fitting successive logistic regressions of X_i on $X_1 \dots X_{i-1}$ using maximum likelihood. a_{ji} for $i \neq j$ will then be the coefficient on X_j while a_{ii} is the intercept of the regression.

2. If one is just working with a desired cross-moment matrix M, the successive rows of A can be fit via Newton-Raphson.

   Let us say that the first $i-1$ rows have already been fit, resulting in the upper left $(i-1) \times (i-1)$ sub matrix A_{-i} of A. Let us say that $\mathbf{a}_i$ is the first i entries of the ith row of A (the rest will be 0 anyway). As well, let $\mathbf{m}_i$ be similarly the first i entries of the ith row of M. In other words, $\mathbf{m}_i = [\mathrm{E}(x_i x_j)]_{j=1}^{i}$. Finally, let us say that $\mathbf{x}_{-i}$ is the first $i-1$ entries of $\mathbf{x}$. We want to solve for $\mathbf{a}_i$ such that

$$\mathbf{m}_i = \mathrm{E}\left(x_i \begin{bmatrix} \mathbf{x}_{-i} \\ x_i \end{bmatrix}\right)$$

$$\mathbf{m}_i = \mathrm{E}\left(\mathrm{E}\left(x_i \begin{bmatrix} \mathbf{x}_{-i} \\ x_i \end{bmatrix} \,\middle|\, \mathbf{x}_{-i}\right)\right)$$

$$\mathbf{m}_i = \sum_{\mathbf{x}_{-i} \in \{0,1\}^{i-1}} \mathrm{P}(\mathbf{x}_{-i}) \mathrm{P}(x_i = 1 \mid \mathbf{x}_{-i}) \begin{bmatrix} \mathbf{x}_{-i} \\ 1 \end{bmatrix}$$

$$\mathbf{m}_i = \sum_{\mathbf{x}_{-i} \in \{0,1\}^{i-1}} \frac{1}{c} \mu\left(\mathbf{x}'_{-i} A_{-i} \mathbf{x}_{-i}\right) \mu\left(\mathbf{a}'_i \begin{bmatrix} \mathbf{x}_{-i} \\ 1 \end{bmatrix}\right) \begin{bmatrix} \mathbf{x}_{-i} \\ 1 \end{bmatrix}$$

   Where c is the appropriate normalizing constant. Let us define the quantity on the right in the last line as $f(\mathbf{a}_i)$. We can solve for $f(\mathbf{a}_i) = \mathbf{m}_i$ by successive Newton-Raphson iterations defined by

$$\mathbf{a}_i^{(k+1)} = \left[Hf\left(\mathbf{a}_i^{(k)}\right)\right]^{-1} \left[f\left(\mathbf{a}_i^{(k)}\right) - \mathbf{m}_i\right]$$

   The Hessian matrix is calculated as

$$Hf(\mathbf{a}_i) = \sum_{\mathbf{x}_{-i} \in \{0,1\}^{i-1}} \frac{1}{c} \mu\left(\mathbf{x}'_{-i} A_{-i} \mathbf{x}_{-i}\right) \mu'\left(\mathbf{a}'_i \begin{bmatrix} \mathbf{x}_{-i} \\ 1 \end{bmatrix}\right) \begin{bmatrix} \mathbf{x}_{-i} \\ 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}'_{-i} & 1 \end{bmatrix}$$

   With 2^{i-1} possible values for $\mathbf{x}_{-i}$, this can quickly become computationally expensive. Instead, with a series of values $\mathbf{x}_{-i}^{(k)} \sim \mathbf{x}_{-i}$, we can approximate

$$f(\mathbf{a}_i) \approx \frac{1}{N} \sum_{k=1}^{N} \mu\left(\mathbf{a}'_i \begin{bmatrix} \mathbf{x}_{-i}^{(k)} \\ 1 \end{bmatrix}\right) \begin{bmatrix} \mathbf{x}_{-i}^{(k)} \\ 1 \end{bmatrix}$$

   and

$$Hf(\mathbf{a}_i) \approx \frac{1}{N} \sum_{k=1}^{N} \mu'\left(\mathbf{a}'_i \begin{bmatrix} \mathbf{x}_{-i}^{(k)} \\ 1 \end{bmatrix}\right) \begin{bmatrix} \mathbf{x}_{-i}^{(k)} \\ 1 \end{bmatrix} \begin{bmatrix} [x_{-i}^{(k)}]' & 1 \end{bmatrix}$$

4 Generating Knockoffs