Conditional Knockoff

This [R] notebook illustrate the method of conditional knockoffs.

```
library(knockoff)
source('knockoff_measure.R')
source('util.R')
```

Multivariate Gaussian Models

In this scenario we assume that the distribution of the covariate belongs to a multivariate Gaussian family with unknown mean and covariance matrix, and we make no assumptions on the conditional distribution of the response. For simplicity, we will use synthetic data constructed from a linear model such that the response only depends on a small fraction of the variables.

Low-Dimensional

We first consider the case when the number of observations n is greater than $2p$.

```
set.seed(2019)
source('cknock_ldg.R')
# Problem parameters
p=400
n=3*p
k=60
A=4
nonzero = sample(p, k)
beta = 1  * (1:p %in% nonzero) * sample(c(-1,1),p,rep=T)
# Generate the variables from a multivariate normal distribution
rho=0.3
Sigma=rho^abs(outer(1:p,1:p,'-'))
Sigma.chol = chol(Sigma)
X = matrix(rnorm(n * p), n)  %*% Sigma.chol

# Generate the response from a linear model
Y=X%*%beta*A/sqrt(n) + rnorm(n)
```

The conditional kncokfoff can is generated by ...

```
Xk.cond=cknockoff.ldg(X,method = 'mix')
```

Then it is routinely used by knockoff filter to select variables.

```
fdr.level = 0.2
knockoff.stat.fun =  mod.stat.lasso_coefdiff
filter.cond=knockoff.filter(X,Y,
        knockoffs = function(x) {
          Xk.cond
        },
        statistic = knockoff.stat.fun,
        fdr = fdr.level,offset = 1)
# false positive and false negative
```

```
c(fp(filter.cond$selected, beta),
  fn(filter.cond$selected, beta))
```

```
## [1] 0.07843137 0.78333333
```

High-Dimensional with Unlabled Data

```
# Problem parameters
n=p/5
n.u=2*p
# Generate covariate
X = matrix(rnorm(n * p), n)  %*% Sigma.chol
X.u = matrix(rnorm(n.u * p), n.u)  %*% Sigma.chol

# Generate the response from a linear model
Y=X%*%beta*A/sqrt(n) + rnorm(n)
```

The conditional kncokfoff can is generated by ...

```
Xk.star=cknockoff.ldg(rbind(X,X.u),method = 'mix')
Xk.cond=Xk.star[1:n,]
```

```
filter.cond=knockoff.filter(X,Y,
        knockoffs = function(x) {
          Xk.cond
        },
        statistic = knockoff.stat.fun,
        fdr = fdr.level,offset = 1)
# false positive and false negative
c(fp(filter.cond$selected, beta),
  fn(filter.cond$selected, beta))
```

```
## [1] 0 0
```