Square Root Rank Estimation

Shira Kritchman

Friday, July 24, 2015

The function 'sqrtFitMSE' tries to feet sqrt law to the eigenvalues distributionclose to the maximal noise eigenvalue. It gets the vector of eigenvalues 'ell', and also a value q. It performs the fitting assuming that the signal rank is q. The function returns the MSE for the best fit it could find.

```
sqrtFitMSE <- function(q, ell, d=10) {
        p <- length(ell)
        i <- p - which.max(ell[p:1]>0) + 1 #index of the last non-zero value at ell
        range <- (ell[q+1] - ell[i]) / d #we want to consider a range which is 1/d from the total range
        j <- which.min(ell > (ell[q+1]-range)) #we want to take eigenvalues that are between ell[q+1] a
        # we should take ell from (q+1) to j
        if (j < q+10) {j <- q+10} #we want to use at least 10 eigenvalues
        x <- ell[(q+1):j]
        u <- ((p-q):(p-j+1))/(p)
        y <- (1-u)^(2/3)
        l <- (ell[q+1] + ell[q+2]) / 2
        if (q==0) {
                L <- Inf
                } else {
                        L <- (ell[q] + ell[q+1]) / 2
                        }
        constrainedFit <- nls(x ~ m + alpha*y, algorithm="port",
                              start=c(m=x[1], alpha=(x[1]-x[2])/(y[1]-y[2])),
                              lower=c(m=l,alpha=-Inf),
                              upper=c(m=L,alpha=Inf)
                              )
        return(mean(resid(constrainedFit)^2))
        }
```

The function 'kEst' gets as input the vector of eigenvalues 'ell'. For each possible value 'q' of the rank (currently limited to q in 0:10) it uses 'sqrtFitMSE' to compute the MSE of fitting a sqrt law to the distribution of the eigenvalues, under the assumption of 'q' signals. It then estimate the rank by taking 'q' with the minimal MSE.

```
kEst <- function(ell, k_max=10) {
        mse <- rep(0,k_max+1)
        for (q in 0:max_k) {
                mse[q+1] <- sqrtFitMSE(q, ell)
                }
        k <- which.min(mse)-1
}
```

Testing the algorithm: we test the estimator with various values for p, where $n = p$, noise eigenvalues are $U[0.5, 1.5]$, and we have rank $k = 2$ with signal strengths $\lambda = (10, 4)$. For each value of p we make 100 iterations and show the fraction of correct estimations.

```
set.seed(265)
P <- 2^(5:9)
lam <- c(10,4)
k <- length(lam)
max_k <- 10
iter <- 100
frac <- rep(0,length(P))
for (i_P in 1:length(P)) {
        p <- P[i_P]
        n <- p
        count <- 0
        for (i_iter in 1:iter) {
                Sigma <- diag(c(lam, rep(0,p-k)) + runif(p,0.5,1.5))
                S <- rWishart(1,n,Sigma)
                S <- S[,,1]/n
                ell <- eigen(S,TRUE,TRUE)$values
                k_hat <- kEst(ell)
                if (k_hat == k) {count <- count+1}
                }
        frac[i_P] <- count/iter
        }
plot(P, frac, ylim=c(0,1), pch=16, main="Probability of Estimating the Correct Number of Signals as func
abline(1, 0, col="red", lwd=3)
```