James Stein Quiz

Naveen Narayanan Meyyappan

04/12/2019

## Question

Compare the risk of the James stein estimator vs the MLE for k =1 to 100 for random thetai, plot it, use 1000 samples at each k to estimate risk., use N(thetai, 1) for the samples. (rnorm(k))

## Answer

library(ggplot2)  
set.seed(1)

The James–Stein estimator is a biased estimator of the mean of Gaussian random vectors. Let us define a function to compute both the James Stein Estimator and MLE estimator. Then we will use 1000 samples to determine the risk at each k (k ranges from 1 to 100). James-Stein estimate is just the sample mean shrunk towards zero.

signcheck<-function(x){ # this function is used to check the sign of the input. Returns zero if the input is negative  
 if(x>0){  
 return(x)  
 }  
 else{  
 return(0)  
 }  
}  
  
computejsrisk<-function(X,mu,n,k){  
 theta\_mle=colMeans(X) # theta for mle is the mean of random normal multivariate distribution  
 theta\_mle\_norm = sqrt(sum(theta\_mle\*theta\_mle)) # We will normalize the theta\_mle vector  
 theta\_js = (signcheck(1-((k-2)/(n\*(theta\_mle\_norm\*theta\_mle\_norm))))\*theta\_mle) # Computing the James Stein Estimator  
 risk\_js = sum((mu - theta\_js)\*(mu - theta\_js)) # Calculating the risk for the James Stein Estimator  
 return(risk\_js)  
}  
computemlerisk<-function(X,mu){  
 theta\_mle=colMeans(X) # theta for mle is the mean of random normal multivariate distribution  
 risk\_mle = sum((mu-theta\_mle)^2) # Calculating the mle\_risk  
 return(risk\_mle)  
}

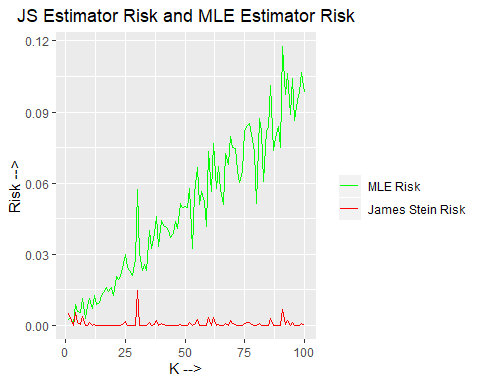
The above two functions are used to compute both the James Stein Risk and MLE Risk.

Now let us simulate the above two functions for n=1000 and k=1:100.

n = 1000 # For 1000 samples  
k=100 # k varies from 1 to 100  
js\_risk = {}  
mle\_risk = {}  
for (i in 1:100){  
 mu=rep(0,i) # Creating a vector that denotes the mean of the normal distribution which is now set as zero  
 val=rnorm((i\*n),0,1) # Creating multivariate normal distribution of size n x k  
 X=matrix(val,nrow=n,ncol=i) # Converting the vector to a matrix  
 mle\_val=computemlerisk(X,mu) # Compute mle\_risk  
 js\_val = computejsrisk(X,mu,n,i) # Compute js\_risk  
 mle\_risk=c(mle\_risk,mle\_val) # Append the results for each k  
 js\_risk=c(js\_risk,js\_val)  
}

The vector mle\_risk and js\_risk are used to store the result. Now we will plot the result to compare the James Stein Estimator and MLE Estimator.

library(ggplot2)  
xaxisvalues=1:100  
result=data.frame(xaxisvalues,mle\_risk,js\_risk)  
ggplot() + geom\_line(data = result, aes(x = result$xaxisvalues, y = result$mle\_risk, colour = "MLE Risk")) + geom\_line(data = result, aes(x = result$xaxisvalues, y = result$js\_risk, color = "James Stein Risk")) + labs(title="JS Estimator Risk and MLE Estimator Risk", x="K -->", y="Risk -->") + theme(plot.title = element\_text(hjust = 0.5)) + scale\_colour\_manual("", breaks = c("MLE Risk", "James Stein Risk"),values = c("red", "green", "blue"))



## References

1. <https://en.wikipedia.org/wiki/James%E2%80%93Stein_estimator>
2. <https://www.naftaliharris.com/blog/steinviz/>