Comment on: “Selective Inference: The Silent Killer of Replicability” by Yoav Benjamini

Erik van Zwet

28 december, 2020

Contents

1 Introduction

This is a comment on an article by Yoav Benjamini entitled “Selective Inference: The Silent Killer of Replicability”, which is available here.

I completely agree with the main point of the article that over-optimism due to selection (a.k.a. the winner’s curse) is a major problem. One important line of defense is to correct for multiple testing, and this is discussed in detail.

In my opinion, another important line of defense is shrinkage, and so I was surprised that the Bayesian approach is dimissed rather quickly. In particular, a blog post by Andrew Gelman is criticized. The post has the provocative title: “Bayesian inference completely solves the multiple comparisons problem” and can be found here.

In his post, Gelman samples “effects” from the N(0,0.5) distribution and observes them with standard normal noise. He demonstrates that the posterior mean and 95% credible intervals continue to perform well under selection.

In section 5.5 of Benjamini’s paper the N(0,0.5) is slightly perturbed by mixing it with N(0,3) with probability 1/1000. As a result, the majority of the credibility intervals that do not cover zero come from the N(0,3) component. Under the N(0,0.5) prior, those intervals get shrunken so much that they miss the true parameter.

It should be noted, however, that those effects are so large that they are very unlikely under the N(0,0.5) prior. Such “data-prior conflict” can be resolved by having a prior with a flat tail. This is a matter of “Bayesian robustness” and goes back to a paper by Dawid which can be found here.

Importantly, this does *not* mean that we need to know the true prior. We can mix the N(0,0.5) with almost any wider normal distribution with almost any probability and then very large effects will hardly be shrunken. Here, I demonstrate this by usin the mixture 0.99*N(0,0.5)+0.01*N(0,6) as prior. This is quite far from the truth, but nevertheless, the posterior inference is quite acceptable. We find that among one million simulations, there are 741 credible intervals that do not cover zero. Among those, the proportion that do not cover the parameter is 0.07 (CI: 0.05 to 0.09). See section "Results" below.

The point is that the procedure merely needs to recognize that a particular observation is unlikely to come from N(0,0.5), and then apply very little shrinkage.

My own views on shrinkage in the context of the winner's curse are here. In particular, a form of Bayesian robustness is discussed in section 3.4 of a preprint of myself and Gelman here.

2 Helper functions

2.1 Normal mixture

We define some functions for working with normal mixtures.

```
dmix = function(x,p,m,s){       # normal mixture density
  p %*% sapply(x, function(x) dnorm(x,mean=m,0,sd=s))
}


rmix = function(n,p,m,s){   # sample from a normal mixture
  d=rmultinom(n,1,p)
  rnorm(n,m%*%d,s%*%d)
}

pmix = function(x,p,m,s){        # cumulative distr
  p %*% sapply(x, function(x) pnorm(x,mean=m,sd=s))
}

rmix = function(n,p,m,s){
  d=rmultinom(n,1,p)
  rnorm(n,m%*%d,s%*%d)
}

qfun = function(q,p,m,s){        # quantile function scalar q
  uniroot(function(x) pmix(x,p,m,s)-q, interval=c(-20,20))$root
}

qmix = function(q,p,m,s){        # quantile function vector q
  sapply(q, function(q) qfun(q,p=p,m=m,s=s) )
}
```

2.2 Posterior

We also define a function that computes the posterior distribution of a parameter with a normal mixture prior which is observed with normal noise.

```
posterior = function(x,s,p,mu,sd){  # compute conditional distr of theta given (x,s)
                                    # mixture distr of theta given by (p,mu,sd)
  s2=s^2
  sd2=sd^2
  q=p*dnorm(x,mu,sqrt(sd2+s2))      # conditional mixing probs
  q=q/sum(q)
  pm=(mu*s2 + x*sd2)/(sd2+s2)       # conditional means
  pv=sd2*s2/(sd2+s2)                # conditional variances
```

```
  ps=sqrt(pv)                        # conditional std devs
  data.frame(q,pm,ps)
}
```

3 Data generation

We generate a million theta's from a normal mixture 0.999 * N(0,0.5) + 0.001 * N(0,3). We add standard normal noise to these theta's to get our observations x.

```
set.seed(1234)
N=10^6
p=c(0.999,0.001)
sd=c(0.5,3)
theta=rmix(N,p,m=c(0,0),sd)
x=theta + rnorm(N,0,1)
```

4 Inference

We compute the posterior distribution of theta, and the 95% credible interval. As our prior, we use the mixture 0.99 * N(0,0.5) + 0.01 * N(0,6)

```
ci=matrix(NA,N,2)
for (i in 1:N){
  post=posterior(x=x[i],s=1,p=c(0.99,0.01),mu=c(0,0),sd=c(0.5,6))
  ci[i,]=qmix(c(0.025,0.975),p=post$q,m=post$pm,s=post$ps)
}

boole1=(ci[,1]>0 | ci[,2]<0)          # do not cover zero
boole2=(ci[,1]>theta | ci[,2]<theta)  # do not cover parameter
test=binom.test(x=sum(boole2[boole1]),n=sum(boole1))
ci=test$conf.int
```

5 Results

We have the following results:

- The proportion of intervals that do not cover zero is 0.001.
- The proportion of intervals that do not cover the parameter is 0.049.
- There are 741 intervals that do not cover zero. Among those, the proportion that do not cover the parameter is 0.067 (CI: 0.05 to 0.088).