Eigenvectors of Symmetric Matrices

Ali Taqi

11/4/2020

Computational Evidence: Real Symmetric Matrices have Real Eigenvectors

In this document, we hope to show that given any arbitrary element of the set of $M \times M$ symmetric matrices, denote it $S \in \mathcal{SM}_{\mathbb{R}}[M \times M]$ has a set of real eigenvectors $[\lambda_i] \in \mathbb{R}^M$.

To simulate a generic element S, we use the following method:

(1) First, pick some $f \in [0, 1]$, letting it denote the fraction of positive entries of S. That is;

$$\text{Want: } f \approx \frac{|s_{ij} > 0|}{M^2}$$

We hope to show that our condition is invariant to the value of f, since there is the possibility that the sign proportions of our matrix S influences the $\det(S)$.

(2) To simulate a symmetric matrix S with a fraction of positive entries f, we will sample from the distribution:

$$s_{ij} \sim \text{Unif}(f - 1, f)$$

(3) To not constrict the sizes of $|s_{ij}|$, we will add an ϵ term and scale our endpoints to preserve the fraction f.

$$s_{ij} \sim \text{Unif}(\epsilon(f - 1), \epsilon f)$$

```
unif_fpos <- function(M,f,ep){
  # unless specifically initialized, a random fraction will be chosen
  if(F){
    f <- runif(1,0,1)
    paste("f: ",f,sep="")
  }
  b <- f
  a <- (f-1)
  dist <- data.frame(x = runif(M**2, ep*a, ep*b))
  dist <- dist %>% mutate(x_neg = ifelse(x < 0,yes = 1, no = 0))
  dist
}
```

(4) Having our uniform distribution, we will generate M^2 entries and insert them in the matrix S. Then, we delete the lower triangular matrix, then duplicate the entries from the upper triangle to the lower triangle.

```
make_symm <- function(dist){
  N <- sqrt(length(dist$x))
  P <- matrix(data = dist$x, nrow = N, ncol = N)
  LT <- lower.tri(P)
```

```
  UT <- upper.tri(P)
  P[LT] <- P[UT]
  P
}
```

(5) Now, if we let $f \sim \text{Unif}(0, 1)$ and let $\epsilon \to \infty$, we can well approximate $S \in \mathcal{SM}_{\mathbb{R}}[M \times M]$.

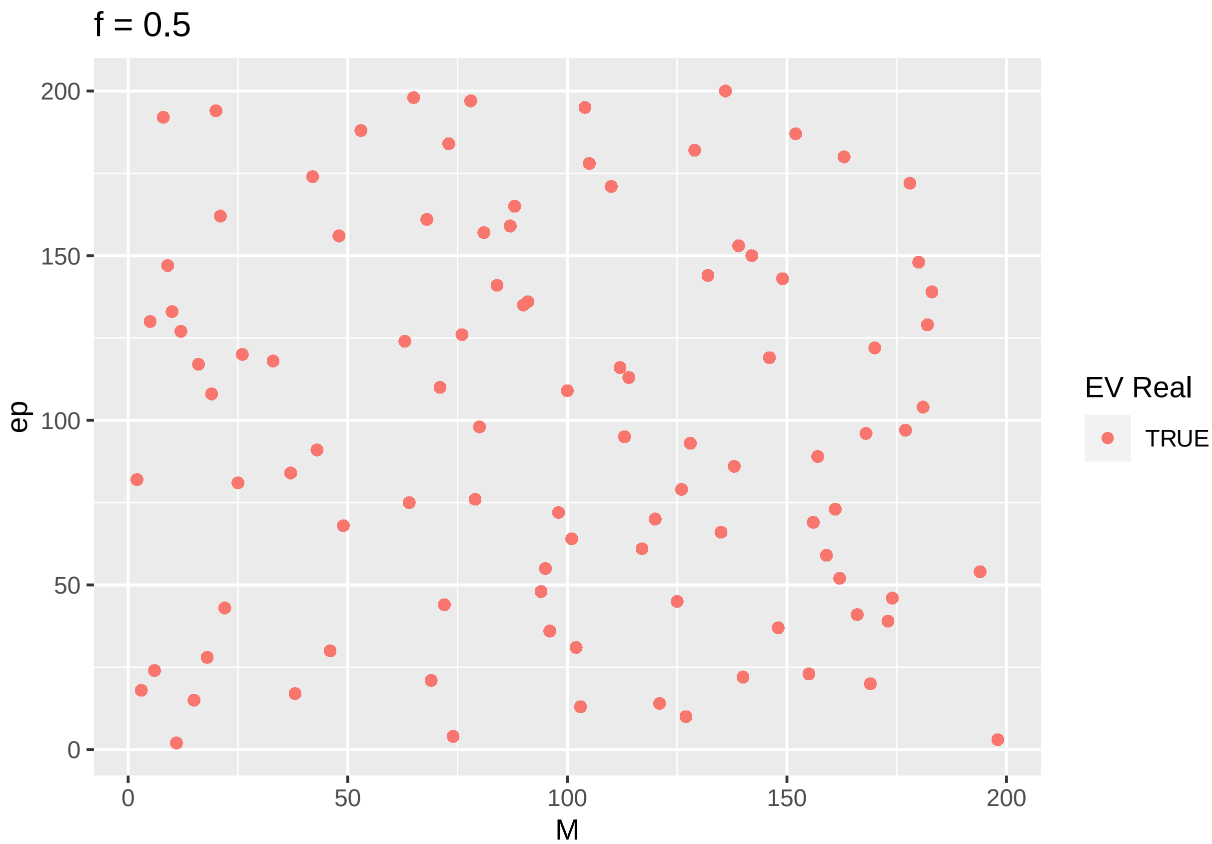
Simulation

```
# add matrix so that we plot(ep,M) on xy plane and color value of diff to show 2d relation of convergen
f <- 0.5

M_vec <- sample(2:200, 100,replace = F)
ep_vec <- sample(2:200, 100,replace = F)
table <- data.frame(M = M_vec, ep = rep(ep_vec,length(M_vec)))

check_real_eigenvectors <- function(M,ep,f){
  S <- RM_symm(M,f,ep)
  S_real <- eigenvectors_real(eigen_frame(S)) # use function for eigenvectors.R
  S_real
}

eigenvecs_real <- rep(NA, length(table$M))

for(i in 1:length(table$M)){
  eigenvecs_real[i] <- check_real_eigenvectors(table[i,][1],table[i,][2],f)
}
table <- cbind(table,eigenvecs_real)
```

```
head(table)
```

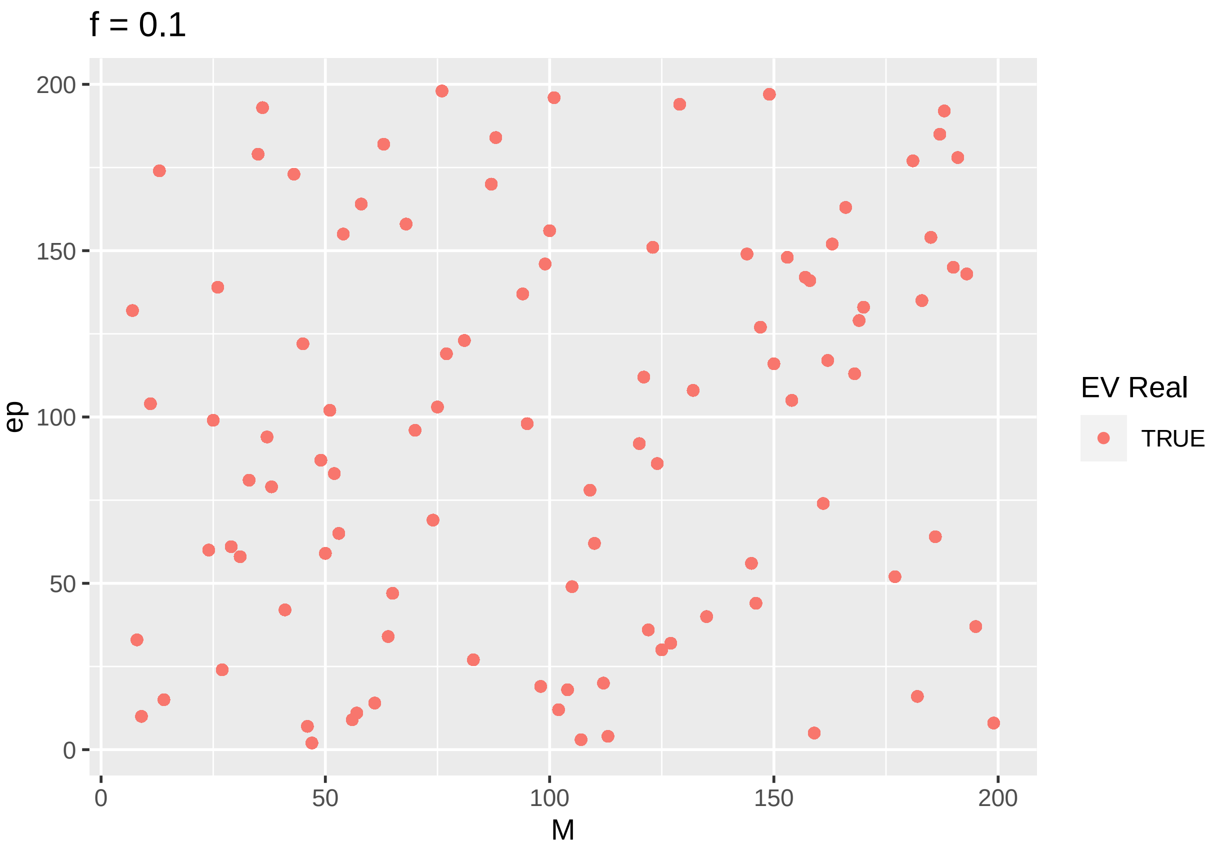
```
##     M  ep eigenvecs_real
## 1 105 178           TRUE
## 2  78 197           TRUE
## 3 125  45           TRUE
## 4  43  91           TRUE
## 5  94  48           TRUE
## 6  87 159           TRUE
```

```
ggplot() +
  geom_point(data = table, aes(x=M,y=ep, color = factor(eigenvecs_real))) +
  labs(color = "EV Real", title = "f = 0.5")
```



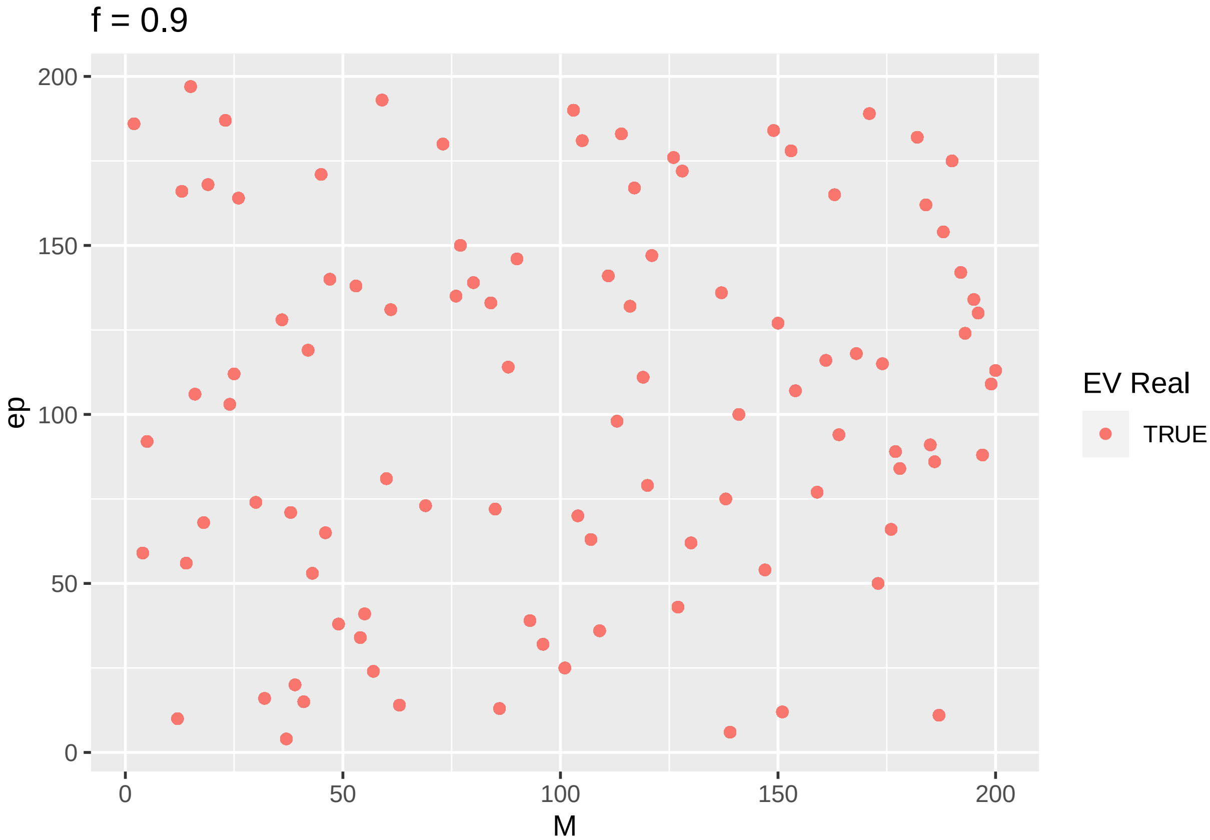

```
# add matrix so that we plot(ep,M) on xy plane and color value of diff to show 2d relation of convergen
f <- 0.1

M_vec <- sample(2:200, 100,replace = F)
ep_vec <- sample(2:200, 100,replace = F)
table <- data.frame(M = M_vec, ep = rep(ep_vec,length(M_vec)))

check_real_eigenvectors <- function(M,ep,f){
  S <- RM_symm(M,f,ep)
  S_real <- eigenvectors_real(eigen_frame(S)) # use function for eigenvectors.R
  S_real
}

eigenvecs_real <- rep(NA, length(table$M))

for(i in 1:length(table$M)){
  eigenvecs_real[i] <- check_real_eigenvectors(table[i,][1],table[i,][2],f)
}
table <- cbind(table,eigenvecs_real)
```

```
ggplot() +
  geom_point(data = table, aes(x=M,y=ep, color = factor(eigenvecs_real))) +
  labs(color = "EV Real", title = "f = 0.1")
```




```
# add matrix so that we plot(ep,M) on xy plane and color value of diff to show 2d relation of convergen
f <- 0.9

M_vec <- sample(2:200, 100,replace = F)
ep_vec <- sample(2:200, 100,replace = F)
table <- data.frame(M = M_vec, ep = rep(ep_vec,length(M_vec)))

check_real_eigenvectors <- function(M,ep,f){
  S <- RM_symm(M,f,ep)
  S_real <- eigenvectors_real(eigen_frame(S)) # use function for eigenvectors.R
  S_real
}

eigenvecs_real <- rep(NA, length(table$M))

for(i in 1:length(table$M)){
  eigenvecs_real[i] <- check_real_eigenvectors(table[i,][1],table[i,][2],f)
}
table <- cbind(table,eigenvecs_real)
```

```
ggplot() +
  geom_point(data = table, aes(x=M,y=ep, color = factor(eigenvecs_real))) +
  labs(color = "EV Real", title = "f = 0.9")
```



f = 0.9
200
150
100
50
0
ep
0
50
100
150
200
M
EV Real
TRUE