

hierarchical_model_v2

Yi Li

08/18/2015

1. Review

Three important elements at location x , time t :

- Larval production ($L_{x,t}$),
- recruitment ($R_{x,t}$),
- adult abundance ($A_{x,t}$),

together with the environment features $E_{x,t}$

Three stages:

- Larval production:

$$L_{x,t} = f(A_{x,t}, E_{x,t}) = p_{x,t} \cdot A_{x,t} \cdot E_{x,t}$$

- Recruitment

$$R_{x,t} = k + \log \left(\sum_{i=1}^N L_{i,t} \cdot e^{-\frac{(d_{x,i} - \mu_d)^2}{\sigma_d^2}} \right) + \beta \cdot E_{x,t} + \epsilon$$

- Adult abundance

$$A_{x,t+1} = b \cdot R_{x,t} \cdot E_{x,t+1}$$

2. Update to second order of the environment feature

2.1 Larval production:

$$\log L_{x,t} = (\log A_{x,t})(\beta_0 + \beta_1 E_{x,t} + \beta_2 E_{x,t}^2)$$

This is equivalent to

$$L_{x,t} = A_{x,t} e^{\beta_0 + \beta_1 E_{x,t} + \beta_2 E_{x,t}^2}$$

2.2 Recruitment

$$\log R_{x,t} = k + \log \left(\sum_{i=1}^N L_{i,t} \cdot e^{-\frac{(d_{x,i} - \mu_d)^2}{\sigma_d^2}} \right) + \beta_3 E_{x,t} + \beta_4 E_{x,t}^2 + \epsilon$$

We don't have data for $L_{x,t}$, but we can plug $L_{x,t}$ (from 2.1) in the above equation:

$$\log R_{x,t} = k + \log \left(\sum_{i=1}^N A_{i,t} \cdot e^{\beta_0 + \beta_1 E_{i,t} + \beta_2 E_{i,t}^2 - \frac{(d_{x,i} - \mu_d)^2}{\sigma_d^2}} \right) + \beta_3 E_{x,t} + \beta_4 E_{x,t}^2 + \epsilon$$

2.3 Adult abundance

$$A_{x,t+1} = R_{x,t} e^{\beta_6 + \beta_7 E_{x,t+1} + \beta_8 E_{x,t+1}^2} + m A_{x,t}$$

3. Two stages model

3.1 Model 1

$$\log R_{x,t} = \log \left(\sum_{i=1}^N A_{i,t} \cdot e^{p_4 + p_5 E_{i,t} - p_6 E_{i,t}^2 + p_7 d_{x,i} - p_8 d_{x,i}^2} \right) + p_1 + p_2 E_{x,t} - p_3 E_{x,t}^2 + \epsilon$$

Note that:

- Unimodality of the environment: coefficient of the second order term is negative
- $\operatorname{argmax}(-ax^2 + bx) = \frac{b}{2a}$ implies E^* occurs at $\frac{p_5}{2p_6}$ and $\frac{p_2}{2p_3}$
- Dispersal parameters: $\mu_d = \frac{p_7}{2p_8}$, $\sigma_d = \frac{1}{\sqrt{p_8}}$

3.2 Adult abundance

$$A_{x,t+1} = R_{x,t} e^{p_2 + p_3 E_{x,t+1} + p_4 E_{x,t+1}^2} + p_1 A_{x,t}$$

Note that:

- This is the growth from t to $t + 1$. may try environment variable $E_{x,t}$ or $E_{x,t+1}$

4. Code and Results

4.1 load the data

```
X=read.csv("~/Dropbox/model/adults.recruits.merged.csv")
feature_name=c("year","lat","lon","sitenum","cover",
               "filter_chla_mean","filter_sst_mean","filter_upw_mean","mytilus.rec")
X=X[feature_name]
```

4.2 derive new variables

- next year abundance: $y = A_{x,t+1}$, target variable

```
nsample=dim(X)[1]
X["ycover"]=NA
for(i in 1:nsample){
  idx=(X[, "sitenum"]==X[i, "sitenum"])&(X[, "year"]==X[i, "year"]+1)
  if(sum(idx)){ # if no next value, will be NA
    X[i, "ycover"]=X[idx, "cover"]
  }
}
```

- next year environment: $E_{x,t+1}$, will be used in model 2

```
X["new_upw"]=NA
X["new_chla"]=NA
X["new_sst"]=NA
for(i in 1:nsample){
  idx=(X[, "sitenum"]==X[i, "sitenum"])&(X[, "year"]==X[i, "year"]+1)
  if(sum(idx)){ # if no next value, will be NA
    X[i, "new_upw"]=X[idx, "filter_upw_mean"]
    X[i, "new_chla"]=X[idx, "filter_chla_mean"]
    X[i, "new_sst"]=X[idx, "filter_sst_mean"]
  }
}
```

Find the complete observations and relabel the site number. Note that we only have one observation for site number 8. And after relabeling, site labels are increasing with latitude.

```
X1=X[complete.cases(X),]

lats=subset(X1,year==2001,select = c(lat))[,1]
lons=subset(X1,year==2001,select = c(lon))[,1]
lats=floor(lats*1000)
lons=floor(lons*1000)
# unique identifier of each site, sorted by lats
# using both lat and lon, because there are two sites have the same lat (but different lon)
latlon=paste0(lats,lons)

for(i in 1:dim(X1)[1]){
  X1[i, "sitenum"]=which(latlon==paste0(floor(X1[i, "lat"]*1000), floor(X1[i, "lon"]*1000)))
}
```

- Target variable in model 1: Recruitment $R_{x,t}$

```
R=X1[, "mytilus.rec"]
R[R==0]=min(R[R!=0])/2
R=log(R)
```

- Target variable in model 2: next year abundance $y = A_{x,t+1}$

```
y=X1[, "ycover"]
y[y==0]=min(y[y!=0])/2 # replace 0 to the minimum positive number
y=log(y) # take log of the species number
```

- Distance Matrix

```
D=dist(X1[X1[, "year"]==2001, "lat"], diag = T, upper = T)
D=as.matrix(D)
for(i in 1:48){
  for(j in 1:48){
    if(i>j){
```

```

    D[i,j]=-D[i,j]
  }
}
}

```

Rescale the environment variables, record the mean and standard deviation.

```

mean_upw=mean(X1[, "filter_upw_mean"])
sd_upw=sd(X1[, "filter_upw_mean"])
X1[, "filter_upw_mean"]=(X1[, "filter_upw_mean"]-mean_upw)/sd_upw

```

4.3 Fit model 1

$$\log R_{x,t} = \log \left(\sum_{i=1}^N A_{i,t} \cdot e^{p_4 + p_5 E_{i,t} - p_6 E_{i,t}^2 + p_7 d_{x,i} - p_8 d_{x,i}^2} \right) + p_1 + p_2 E_{x,t} - p_3 E_{x,t}^2 + \epsilon$$

The input of the (maximum) likelihood function, which is equivalent to least square, are

- eight parameters
- data
- target recruitment
- distance matrix

```

milogl=function(p,X1,R,D){
  n=dim(X1)[1]
  f=rep(0,n)
  for(iobs in 1:n){
    this_ker=0
    this_site=X1[iobs,"sitenum"]
    this_year=X1[iobs,"year"]
    # all the sites that have the same year as the current observation
    Xtemp=subset(X1,year==this_year)
    for(j in dim(Xtemp)[1]){
      sitej=Xtemp[j,"sitenum"]
      this_ker=this_ker+Xtemp[j,"cover"]*
        exp(p[4]+p[5]*Xtemp[j,"filter_upw_mean"]-p[6]*Xtemp[j,"filter_upw_mean"]^2
          +p[7]*D[this_site,sitej]-p[8]*D[this_site,sitej]^2)
    }
    #print(this_ker)
    f[iobs]=log(this_ker)+p[1]+p[2]*X1[iobs,"filter_upw_mean"]-p[3]*X1[iobs,"filter_upw_mean"]^2
  }
  return(sum((R-f)^2))
}

```

```

p0=c(0,1,1,0,1,1,1,1)/10
resm1=nlm(milogl,p0,hessian=T,print.level=0,X1=X1,R=R,D=D,iterlim=1e4,steptol=1e-5)

```

```

## Warning in nlm(milogl, p0, hessian = T, print.level = 0, X1 = X1, R = R, :
## NA/Inf replaced by maximum positive value

```

```
## Warning in nlm(m1logl, p0, hessian = T, print.level = 0, X1 = X1, R = R, :
## NA/Inf replaced by maximum positive value
```

```
## Warning in nlm(m1logl, p0, hessian = T, print.level = 0, X1 = X1, R = R, :
## NA/Inf replaced by maximum positive value
```

```
p1=resm1$estimate
p1
```

```
## [1] -0.318243044 -1.412333675 0.343443604 -0.318243044 0.510635608
## [6] 0.522846783 0.089335863 0.004993183
```

```
mu_hat=p1[7]/(2*p1[8])
sigma_hat=1/sqrt(p1[8])
E_in=p1[5]/(2*p1[6])
E_out=p1[2]/(2*p1[3])
mu_hat
```

```
## [1] 8.945783
```

```
sigma_hat
```

```
## [1] 14.15179
```

```
E_in*sd_upw+mean_upw
```

```
## [1] 61.09286
```

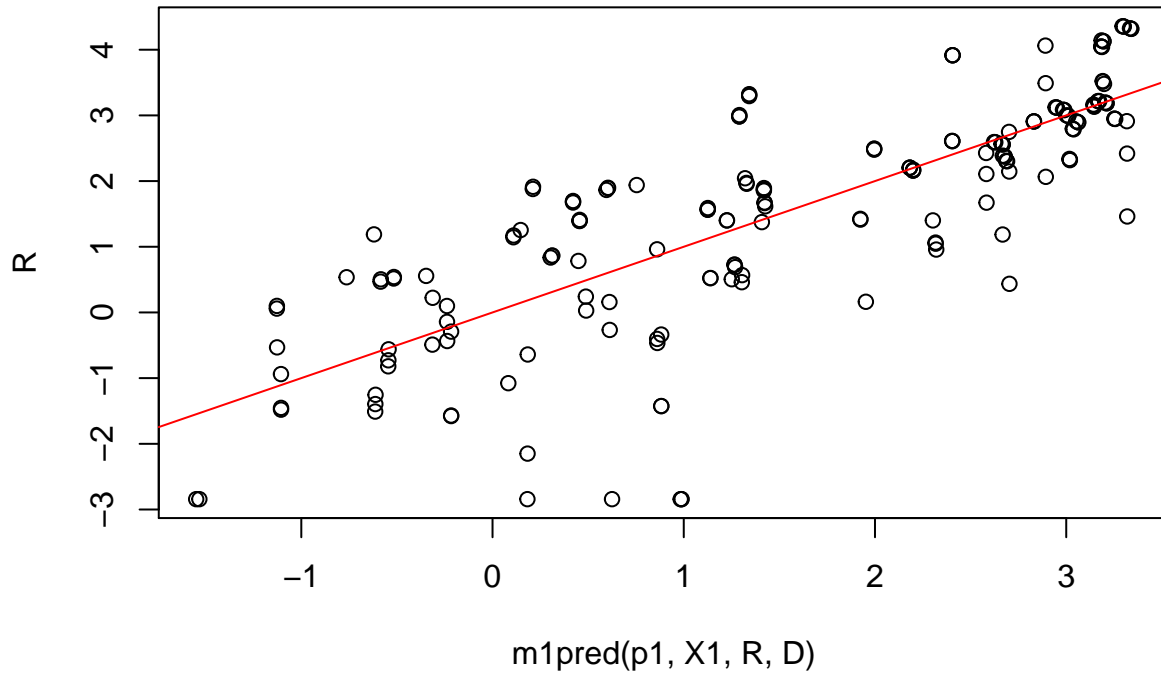
```
E_out*sd_upw+mean_upw
```

```
## [1] -52.27822
```

```
m1pred=function(p,X1,R,D){
  n=dim(X1)[1]
  f=rep(0,n)
  for(iobs in 1:n){
    this_ker=0
    this_site=X1[iobs,"sitenum"]
    this_year=X1[iobs,"year"]
    Xtemp=subset(X1,year==this_year)# all the sites that have the same year as the current observation
    for(j in dim(Xtemp)[1]){
      sitej=Xtemp[j,"sitenum"]
      this_ker=this_ker+Xtemp[j,"cover"]*
        exp(p[4]+p[5]*Xtemp[j,"filter_upw_mean"]-p[6]*Xtemp[j,"filter_upw_mean"]^2
          +p[7]*D[this_site,sitej]-p[8]*D[this_site,sitej]^2)
    }
    #print(this_ker)
    f[iobs]=log(this_ker)+p[1]+p[2]*X1[iobs,"filter_upw_mean"]-p[3]*X1[iobs,"filter_upw_mean"]^2
  }
  return(f)
}
```

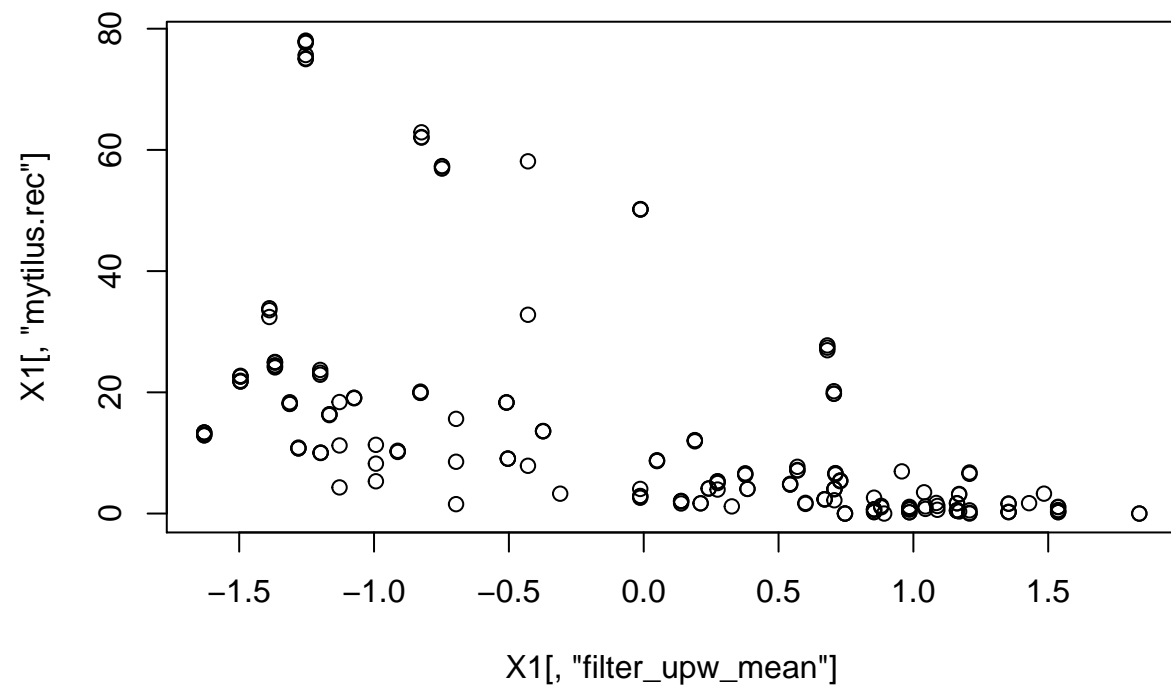
Plot the predicted value and the true value, for model 1(recruitment).

```
plot(m1pred(p1,X1,R,D),R)
abline(a=0,b=1,col=2)
```



Relationship between the environment variable and the recruitment, seem decreasing. Thus, I tried several starting points, t_2 is always negative.

```
plot(X1[, "filter_upw_mean"], X1[, "mytilus.rec"])
```



4.4 Fit model 2

$$A_{x,t+1} = R_{x,t} e^{p_2 + p_3 E_{x,t+1} + p_4 E_{x,t+1}^2} + p_1 A_{x,t}$$

```
m2logl=function(p,X1,y){
  n=dim(X1)[1]
  f=rep(0,n)
  for(iobs in 1:n){
    f[iobs]=X1[iobs,"mytilus.rec"]*exp(p[2]+p[3]*X1[iobs,"new_upw"]-p[4]*X1[iobs,"new_upw"]^2)+p[1]*X1[iobs,"new_upw"]
  }
  return(sum((exp(y)-f)^2))
}
```

```
p0=c(1,0,1,1)
resm2=nlm(m2logl,p0,hessian=T,print.level=0,X1=X1,y=y,iterlim=1e4,steptol=1e-5)
```

```
## Warning in nlm(m2logl, p0, hessian = T, print.level = 0, X1 = X1, y = y, :
## NA/Inf replaced by maximum positive value
```

```
## Warning in nlm(m2logl, p0, hessian = T, print.level = 0, X1 = X1, y = y, :
## NA/Inf replaced by maximum positive value
```

```
## Warning in nlm(m2logl, p0, hessian = T, print.level = 0, X1 = X1, y = y, :
## NA/Inf replaced by maximum positive value
```

```
## Warning in nlm(m2logl, p0, hessian = T, print.level = 0, X1 = X1, y = y, :
## NA/Inf replaced by maximum positive value
```

```
p2=resm2$estimate
p2
```

```
## [1] 0.9134131 -0.1710083 0.1557039 0.2034289
```

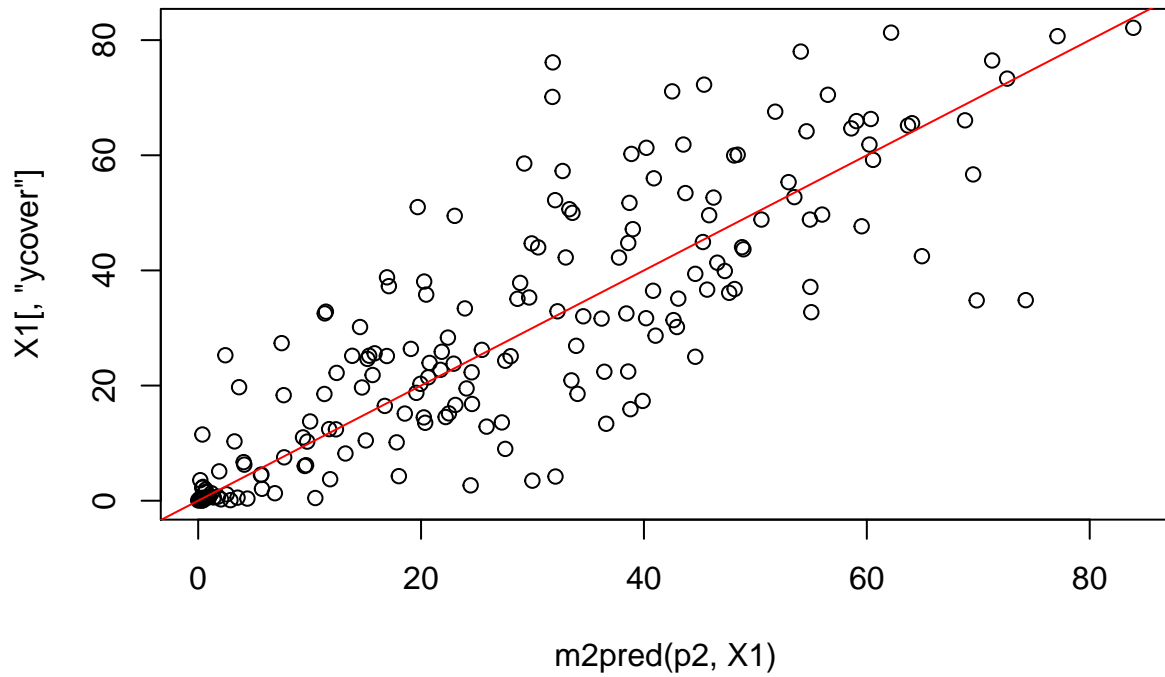
```
E_hat=p2[3]/(2*p2[4])
E_hat*sd_upw+mean_upw
```

```
## [1] 56.38667
```

```
m2pred=function(p,X1){
  n=dim(X1)[1]
  f=rep(0,n)
  for(iobs in 1:n){
    f[iobs]=X1[iobs,"mytilus.rec"]*exp(p[2]+p[3]*X1[iobs,"new_upw"]-p[4]*X1[iobs,"new_upw"]^2)+p[1]*X1[iobs,"new_upw"]
  }
  return(f)
}
```

Plot the predicted value and the true value, for model 2(next year abundance).

```
plot(m2pred(p2,X1),X1[, "ycover"])
abline(a=0,b=1,col=2)
```



5. A simplified model

If ignore the environment effect in the first Larval production step, and self-increment term $mA_{x,t}$ in the last Adult abundance step, then the model is equivalent to the previous model with second order environment terms, i.e.

$$y = K + \log\left(\sum_{i=1}^{n_s} N_{ti} e^{-\frac{(d_{si} - \mu_d)^2}{\sigma_d^2}}\right) + \beta_1 e_{chi} + \beta_2 e_{sst} + \beta_3 e_{upw} + \beta_4 e_{chi}^2 + \beta_5 e_{sst}^2 + \beta_6 e_{upw}^2 + \epsilon,$$

```
library(synchrony)
```

```
## synchrony 0.2.3 loaded.
```

```
data(pisco.data)
y=subset(pisco.data,year>2000,select=c(mussel_abund)) # species number
y[y[,1]==0,]=min(y[y[,1]!=0,])/2 # replace 0 to the minimum positive number
y=log(y) # take log of the species number

N=subset(pisco.data,year<2003,select=c(mussel_abund)) # (past) species number
D=dist(pisco.data[1:48,1],diag = T,upper = T)
D=as.matrix(D)
for(i in 1:48){
  for(j in 1:48){
    if(i>j){
      D[i,j]=-D[i,j]
```



```

    }
  }
}
E=subset(pisco.data,year<2003,select=c(chl,sst,upwelling)) # Environment variable
D=as.matrix(D)

logl2=function(t,y,N,D,E){
  n=dim(y)[1]
  f=rep(0,n) # value of the regression function
  yr=rep(0:2,each=48) # 3 years(0-2) used: 00-03 for X, 01-04 for y
  for(j in 1:n){
    Ker=exp(-(D[j%48+((j%48)==0)*48,]-t[2])^2/(t[3]^2)) # dispersal kernal (for the 48 sites)
    Nj=N[(yr[j]*48+1):((yr[j]+1)*48),] # number of species for 48 sites
    f[j]=t[1]+log(sum(Ker*Nj))+
      t[4]*E[j,1]+t[5]*E[j,2]+t[6]*E[j,3]+
      t[7]*E[j,1]^2+t[8]*E[j,2]^2+t[9]*E[j,3]^2 # dispersal + three environment terms
  }

  return(sum((y[,1]-f)^2)) # return the objective function
}

t0=c(1,0,1,10,10,20,10,10,20)
res2=nlm(logl2,t0,hessian=T,print.level=1,y=y,N=N,D=D,E=E,iterlim=1e4,steptol=1e-5)

```

```

## iteration = 0
## Step:
## [1] 0 0 0 0 0 0 0 0 0
## Parameter:
## [1] 1 0 1 10 10 20 10 10 20
## Function Value
## [1] 2.122045e+12
## Gradient:
## [1] 27491699 -1758301 25335693 78249487 361077832
## [6] 2072565820 374293237 4858458643 207291276196
##
## iteration = 114
## Parameter:
## [1] -0.4573546269 1.3175969823 2.3590791104 0.3587083205 -0.1164384755
## [6] 0.0126321397 -0.0401453764 -0.0094495968 -0.0001124905
## Function Value
## [1] 305.9591
## Gradient:
## [1] 0.49101322 7.51581734 1.61753681 0.24373287 0.97208249 0.03547507
## [7] -0.04966438 -0.09805535 0.06221489
##
## Successive iterates within tolerance.
## Current iterate is probably solution.

```

```

t1=res2$estimate
t1

```

```

## [1] -0.4573546269 1.3175969823 2.3590791104 0.3587083205 -0.1164384755
## [6] 0.0126321397 -0.0401453764 -0.0094495968 -0.0001124905

```

```
t1[2]
```

```
## [1] 1.317597
```

```
t1[3]
```

```
## [1] 2.359079
```

```
-t1[6]/(2*t1[9])
```

```
## [1] 56.14757
```

For the result parameter t , note that $\mu_d = t_2$, $\sigma_d = t_3$, $E^* = -\frac{t_6}{2t_9}$.