

Discrete-Space Continuous-time Models

Devin S. Johnson

NOAA Fisheries

Pacific Islands Fisheries Science Center

Honolulu, Hawaii

Email: devin.johnson@noaa.gov

ISEC2022 Workshop on Tidy Animal Movement
June 25, 2022



NOAA
FISHERIES

Discrete-space and continuous-time?

Do we really care about a 3m movement in 2s?

Usually trying to relate use of a habitat that is measured on a, say 10-40km grid

Trying to model movement in continuous space/time relative to a raster grid of habitat can provide poor results

Why not try to describe movement relative to the observed habitat, raster cells? Or, even spatially separate patches?

Discrete-space and continuous-time?

Do we really care about a 3m movement in 2s?

Usually trying to relate use of a habitat that is measured on a, say 10-40km grid

Trying to model movement in continuous space/time relative to a raster grid of habitat can provide poor results

Why not try to describe movement relative to the observed habitat, raster cells? Or, even spatially separate patches?

Discrete-space and continuous-time?

Do we really care about a 3m movement in 2s?

Usually trying to relate use of a habitat that is measured on a, say 10-40km grid

Trying to model movement in continuous space/time relative to a raster grid of habitat can provide poor results

Why not try to describe movement relative to the observed habitat, raster cells? Or, even spatially separate patches?

Discrete-space and continuous-time?

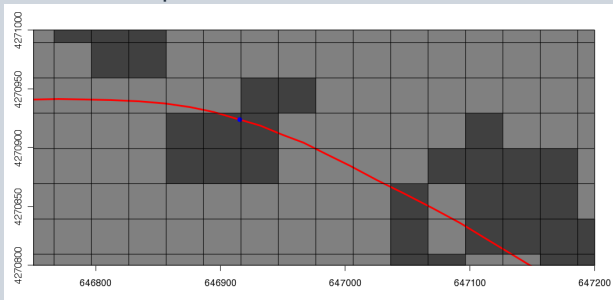
Do we really care about a 3m movement in 2s?

Usually trying to relate use of a habitat that is measured on a, say 10-40km grid

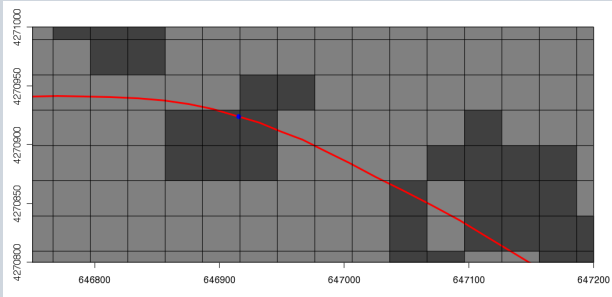
Trying to model movement in continuous space/time relative to a raster grid of habitat can provide poor results

Why not try to describe movement relative to the observed habitat, raster cells? Or, even spatially separate patches?

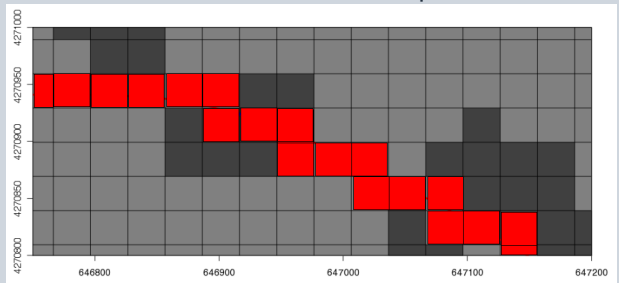
Continuous-space movement:



Continuous-space movement:



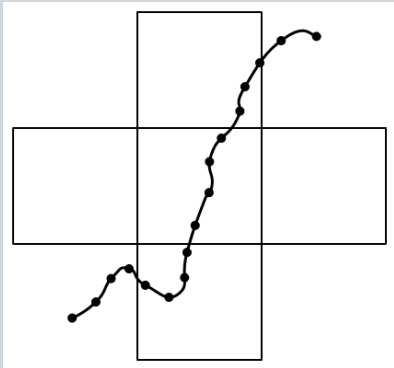
Discrete-space movement:



Continuous-time \rightarrow just make the time steps smaller and smaller

Movement path \mathcal{P} :

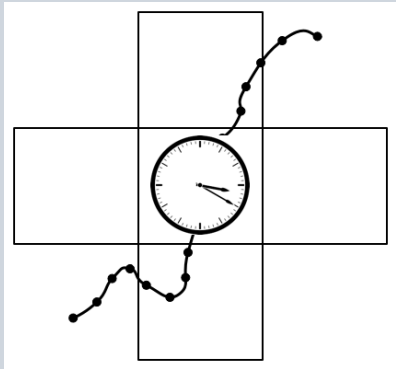
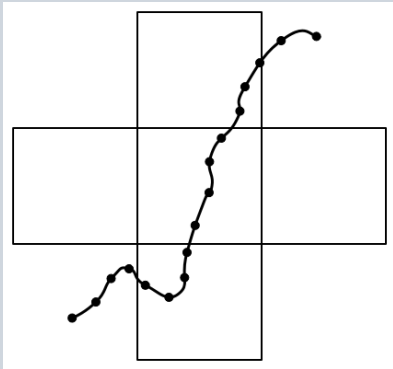
- ① Cells visited
- ② Time stayed in each cell



Continuous-time \rightarrow just make the time steps smaller and smaller

Movement path \mathcal{P} :

- 1 Cells visited
- 2 Time stayed in each cell



Some mathematics

The Notation

Study area is partitioned into discrete “cells” $\mathcal{G} = \{1, \dots, n\}$

animals move from cell, i to cell j **only if** j is a “neighbor” of i ,

$$\text{i.e., } j \in \mathcal{N}_i = \{j \in \mathcal{G} : i \sim j\}$$

Transition times $\{\tau_1, \dots, \tau_K\}$

Residence time $\tau_{k+1} = r_{k+1} + \tau_k$

Embedded chain visited cells, G_1, \dots, G_k

Path $\mathcal{P} = [\tau_0, \mathbf{r}, \mathbf{G}]$

The Notation

Study area is partitioned into discrete “cells” $\mathcal{G} = \{1, \dots, n\}$

animals move from cell, i to cell j **only if** j is a “neighbor” of i ,

$$\text{i.e., } j \in \mathcal{N}_i = \{j \in \mathcal{G} : i \sim j\}$$

Transition times $\{\tau_1, \dots, \tau_K\}$

Residence time $\tau_{k+1} = r_{k+1} + \tau_k$

Embedded chain visited cells, G_1, \dots, G_k

Path $\mathcal{P} = [\tau_0, \mathbf{r}, \mathbf{G}]$

The Notation

Study area is partitioned into discrete “cells” $\mathcal{G} = \{1, \dots, n\}$

animals move from cell, i to cell j **only if** j is a “neighbor” of i ,

$$\text{i.e., } j \in \mathcal{N}_i = \{j \in \mathcal{G} : i \sim j\}$$

Transition times $\{\tau_1, \dots, \tau_K\}$

Residence time $\tau_{k+1} = r_{k+1} + \tau_k$

Embedded chain visited cells, G_1, \dots, G_k

Path $\mathcal{P} = [\tau_0, \mathbf{r}, \mathbf{G}]$

The Notation

Study area is partitioned into discrete “cells” $\mathcal{G} = \{1, \dots, n\}$
animals move from cell, i to cell j **only if** j is a “neighbor” of i ,

$$\text{i.e., } j \in \mathcal{N}_i = \{j \in \mathcal{G} : i \sim j\}$$

Transition times $\{\tau_1, \dots, \tau_K\}$

Residence time $\tau_{k+1} = r_{k+1} + \tau_k$

Embedded chain visited cells, G_1, \dots, G_k

Path $\mathcal{P} = [\tau_0, \mathbf{r}, \mathbf{G}]$

The Notation

Study area is partitioned into discrete “cells” $\mathcal{G} = \{1, \dots, n\}$
animals move from cell, i to cell j **only if** j is a “neighbor” of i ,

$$\text{i.e., } j \in \mathcal{N}_i = \{j \in \mathcal{G} : i \sim j\}$$

Transition times $\{\tau_1, \dots, \tau_K\}$

Residence time $\tau_{k+1} = r_{k+1} + \tau_k$

Embedded chain visited cells, G_1, \dots, G_k

Path $\mathcal{P} = [\tau_0, \mathbf{r}, \mathbf{G}]$

The Notation

Study area is partitioned into discrete “cells” $\mathcal{G} = \{1, \dots, n\}$
animals move from cell, i to cell j **only if** j is a “neighbor” of i ,

$$\text{i.e., } j \in \mathcal{N}_i = \{j \in \mathcal{G} : i \sim j\}$$

Transition times $\{\tau_1, \dots, \tau_K\}$

Residence time $\tau_{k+1} = r_{k+1} + \tau_k$

Embedded chain visited cells, G_1, \dots, G_k

Path $\mathcal{P} = [\tau_0, \mathbf{r}, \mathbf{G}]$

Let's make a model!

Spoiler Alert! \mathcal{P} is a continuous-time Markov chain!

The Process: $G(t)$ is the occupied cell for any $t \in [0, T]$, i.e.,
 $G(t) = G_k$ for any $t \in [\tau_k, \tau_{k+1})$ $G(t)$ is a discrete-state,
continuous-time process!

Some sensible rules:

Let's make a model!

Spoiler Alert! \mathcal{P} is a continuous-time Markov chain!

The Process: $G(t)$ is the occupied cell for any $t \in [0, T]$, i.e.,
 $G(t) = G_k$ for any $t \in [\tau_k, \tau_{k+1})$ $G(t)$ is a discrete-state,
continuous-time process!

Some sensible rules:

Let's make a model!

Spoiler Alert! \mathcal{P} is a continuous-time Markov chain!

The Process: $G(t)$ is the occupied cell for any $t \in [0, T]$, i.e.,
 $G(t) = G_k$ for any $t \in [\tau_k, \tau_{k+1})$ $G(t)$ is a discrete-state,
continuous-time process!

Some sensible rules:

- 1 $[G(t+r) = j | G(t) = i] \approx r\lambda_{ij}$
(The probability goes to 0, but not too fast!)
- 2 $[\geq 2 \text{ transitions before } t+r | G(t) = i] \approx 0$
(Can't have transitions on top of one another!)
- 3 $[0 \text{ transitions before } t+r | G(t) = i] \approx 1 - r \sum_{j \in \mathcal{N}_i} \lambda_{ij}$
(Moves are mutually exclusive)

Let's make a model!

Spoiler Alert! \mathcal{P} is a continuous-time Markov chain!

The Process: $G(t)$ is the occupied cell for any $t \in [0, T]$, i.e.,
 $G(t) = G_k$ for any $t \in [\tau_k, \tau_{k+1})$ $G(t)$ is a discrete-state,
continuous-time process!

Some sensible rules:

- ① $[G(t+r) = j | G(t) = i] \approx r\lambda_{ij}$
(The probability goes to 0, but not too fast!)
- ② $[\geq 2 \text{ transitions before } t+r | G(t) = i] \approx 0$
(Can't have transitions on top of one another!)
- ③ $[0 \text{ transitions before } t+r | G(t) = i] \approx 1 - r \sum_{j \in \mathcal{N}_i} \lambda_{ij}$
(Moves are mutually exclusive)

Let's make a model!

Spoiler Alert! \mathcal{P} is a continuous-time Markov chain!

The Process: $G(t)$ is the occupied cell for any $t \in [0, T]$, i.e.,
 $G(t) = G_k$ for any $t \in [\tau_k, \tau_{k+1})$ $G(t)$ is a discrete-state,
continuous-time process!

Some sensible rules:

- 1 $[G(t+r) = j | G(t) = i] \approx r\lambda_{ij}$
(The probability goes to 0, but not too fast!)
- 2 $[\geq 2 \text{ transitions before } t+r | G(t) = i] \approx 0$
(Can't have transitions on top of one another!)
- 3 $[0 \text{ transitions before } t+r | G(t) = i] \approx 1 - r \sum_{j \in \mathcal{N}_i} \lambda_{ij}$
(Moves are mutually exclusive)

Let's make a model!

Spoiler Alert! \mathcal{P} is a continuous-time Markov chain!

The Process: $G(t)$ is the occupied cell for any $t \in [0, T]$, i.e.,
 $G(t) = G_k$ for any $t \in [\tau_k, \tau_{k+1})$ $G(t)$ is a discrete-state,
continuous-time process!

Some sensible rules:

- 1 $[G(t+r) = j | G(t) = i] \approx r\lambda_{ij}$
(The probability goes to 0, but not too fast!)
- 2 $[\geq 2 \text{ transitions before } t+r | G(t) = i] \approx 0$
(Can't have transitions on top of one another!)
- 3 $[0 \text{ transitions before } t+r | G(t) = i] \approx 1 - r \sum_{j \in \mathcal{N}_i} \lambda_{ij}$
(Moves are mutually exclusive)

Let's make a model!

First: $\Lambda_i = \sum_{j \in \mathcal{N}_i} \lambda_{ij}$ is the total rate of movement when in cell i

Let's make a model!

First: $\Lambda_i = \sum_{j \in \mathcal{N}_i} \lambda_{ij}$ is the total rate of movement when in cell i

Some useful properties:

- 1 Prob. animal moves to j given a move is made at time t :

$$\begin{aligned} [G(t) = j | G(t-r) = i, \text{ move at } t] &\approx \frac{r\lambda_{ij}}{r\Lambda_i} \\ &\rightarrow \frac{\lambda_{ij}}{\Lambda_i} \end{aligned}$$

- 2 $[r | G(t) = i] = \text{Exponential}(\Lambda_i)$

Let's make a model!

First: $\Lambda_i = \sum_{j \in \mathcal{N}_i} \lambda_{ij}$ is the total rate of movement when in cell i

Some useful properties:

- 1 Prob. animal moves to j given a move is made at time t :

$$\begin{aligned} [G(t) = j | G(t-r) = i, \text{ move at } t] &\approx \frac{r\lambda_{ij}}{r\Lambda_i} \\ &\rightarrow \frac{\lambda_{ij}}{\Lambda_i} \end{aligned}$$

- 2 $[r | G(t) = i] = \text{Exponential}(\Lambda_i)$

Let's make a model!

First: $\Lambda_i = \sum_{j \in \mathcal{N}_i} \lambda_{ij}$ is the total rate of movement when in cell i

Some useful properties:

- 1 Prob. animal moves to j given a move is made at time t :

$$\begin{aligned} [G(t) = j | G(t-r) = i, \text{ move at } t] &\approx \frac{r\lambda_{ij}}{r\Lambda_i} \\ &\rightarrow \frac{\lambda_{ij}}{\Lambda_i} \end{aligned}$$

- 2 $[r | G(t) = i] = \text{Exponential}(\Lambda_i)$

Let's make a model!

First: $\Lambda_i = \sum_{j \in \mathcal{N}_i} \lambda_{ij}$ is the total rate of movement when in cell i

Some useful properties:

- 1 Prob. animal moves to j given a move is made at time t :

$$\begin{aligned} [G(t) = j | G(t-r) = i, \text{ move at } t] &\approx \frac{r\lambda_{ij}}{r\Lambda_i} \\ &\rightarrow \frac{\lambda_{ij}}{\Lambda_i} \end{aligned}$$

- 2 $[r | G(t) = i] = \text{Exponential}(\Lambda_i)$

... Wait, what ??? (don't worry it's a CTMC property)

Enough Math! Let's fit a model

The $\mathcal{P} = \{\tau_0 = 0, \mathbf{r}, \mathbf{G}\}$ is really a bivariate time-series

$Z_{kj} = 1$ if $j = G_k$ and 0 else.

Likelihood is Poisson with 'data' $\{Z_{kj}\}$ and rates $\{r_k \lambda_{G_{k-1},j}\}$!!!

Enough Math! Let's fit a model

The $\mathcal{P} = \{\tau_0 = 0, \mathbf{r}, \mathbf{G}\}$ is really a bivariate time-series

The Likelihood

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &= \prod_{k=1}^K [G_k \mid r_k, G_{k-1}] [r_k \mid G_{k-1}] \\ &\propto \prod_{k=1}^K \prod_{j \in \mathcal{N}_{G_{k-1}}} (r_k \lambda_{G_{k-1},j})^{Z_{kj}} \exp\{-r_k \lambda_{G_{k-1},j}\}\end{aligned}$$

$Z_{kj} = 1$ if $j = G_k$ and 0 else.

Likelihood is Poisson with 'data' $\{Z_{kj}\}$ and rates $\{r_k \lambda_{G_{k-1},j}\}$!!!

Enough Math! Let's fit a model

The $\mathcal{P} = \{\tau_0 = 0, \mathbf{r}, \mathbf{G}\}$ is really a bivariate time-series

The Likelihood

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &= \prod_{k=1}^K [G_k \mid r_k, G_{k-1}] [r_k \mid G_{k-1}] \\ &\propto \prod_{k=1}^K \prod_{j \in \mathcal{N}_{G_{k-1}}} (r_k \lambda_{G_{k-1},j})^{Z_{kj}} \exp\{-r_k \lambda_{G_{k-1},j}\}\end{aligned}$$

$Z_{kj} = 1$ if $j = G_k$ and 0 else.

Likelihood is Poisson with 'data' $\{Z_{kj}\}$ and rates $\{r_k \lambda_{G_{k-1},j}\}$!!!

Software and fitting

Log linear model

$$\log r_k \lambda_{ij} = \log(r_k) + \mathbf{m}_i' \boldsymbol{\delta}(\tau_k) + \mathbf{x}_{ij}' \boldsymbol{\gamma}(\tau_k)$$

- \mathbf{m}_i are “motility” covariates that control the tendency to stick in cell i once you arrive (i.e., *“I don’t know what’s around me, but I like it here”*)
- \mathbf{x}_{ij} are “gradient” covariates that draws the animal to cell j once it arrives in i (i.e., *“I can see it is better next door, I’m headed over there now!”*)

Best part?... **The model is linear in the parameters, so use whatever GLM-like software you want!!** Just remember to use $\log(r_k)$ as an offset!

Software and fitting

Log linear model

$$\log r_k \lambda_{ij} = \log(r_k) + \mathbf{m}_i' \boldsymbol{\delta}(\tau_k) + \mathbf{x}_{ij}' \boldsymbol{\gamma}(\tau_k)$$

- \mathbf{m}_i are “motility” covariates that control the tendency to stick in cell i once you arrive (i.e., *“I don’t know what’s around me, but I like it here”*)
- \mathbf{x}_{ij} are “gradient” covariates that draws the animal to cell j once it arrives in i (i.e., *“I can see it is better next door, I’m headed over there now!”*)

Best part?... **The model is linear in the parameters, so use whatever GLM-like software you want!!** Just remember to use $\log(r_k)$ as an offset!

Software and fitting

Log linear model

$$\log r_k \lambda_{ij} = \log(r_k) + \mathbf{m}_i' \boldsymbol{\delta}(\tau_k) + \mathbf{x}_{ij}' \boldsymbol{\gamma}(\tau_k)$$

- \mathbf{m}_i are “motility” covariates that control the tendency to stick in cell i once you arrive (i.e., *“I don’t know what’s around me, but I like it here”*)
- \mathbf{x}_{ij} are “gradient” covariates that draws the animal to cell j once it arrives in i (i.e., *“I can see it is better next door, I’m headed over there now!”*)

Best part?... The model is linear in the parameters, so use whatever GLM-like software you want!! Just remember to use $\log(r_k)$ as an offset!

Software and fitting

Log linear model

$$\log r_k \lambda_{ij} = \log(r_k) + \mathbf{m}_i' \boldsymbol{\delta}(\tau_k) + \mathbf{x}_{ij}' \boldsymbol{\gamma}(\tau_k)$$

- \mathbf{m}_i are “motility” covariates that control the tendency to stick in cell i once you arrive (i.e., *“I don’t know what’s around me, but I like it here”*)
- \mathbf{x}_{ij} are “gradient” covariates that draws the animal to cell j once it arrives in i (i.e., *“I can see it is better next door, I’m headed over there now!”*)

Best part?... **The model is linear in the parameters, so use whatever GLM-like software you want!!** Just remember to use $\log(r_k)$ as an offset!

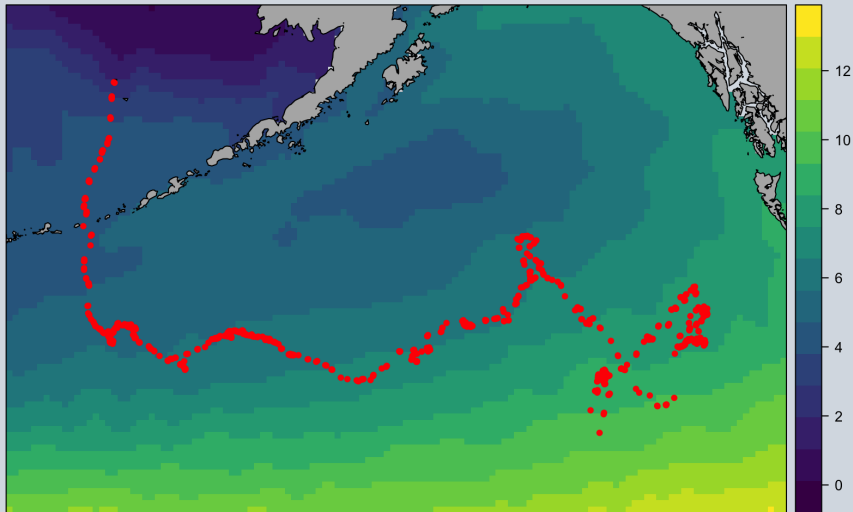
Some data analysis

Spatial covariates and NFS migration



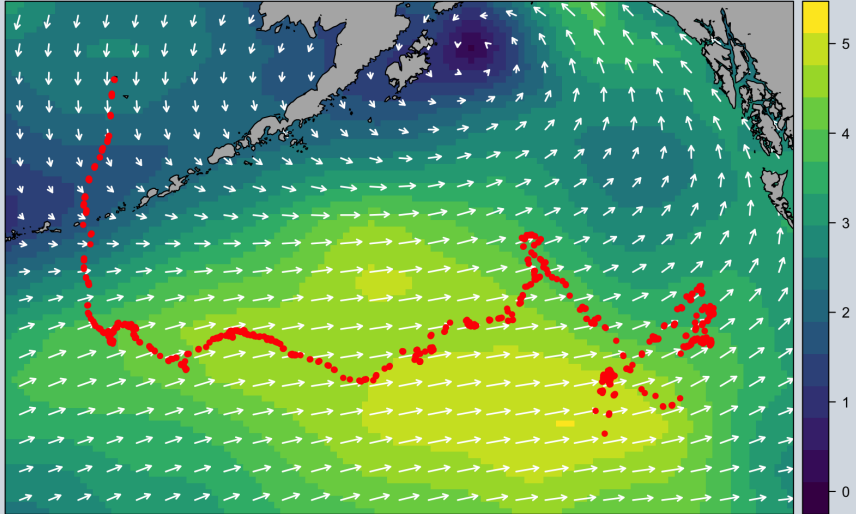
Spatial habitat for NFS migration

Mean Sea Surface Temperature



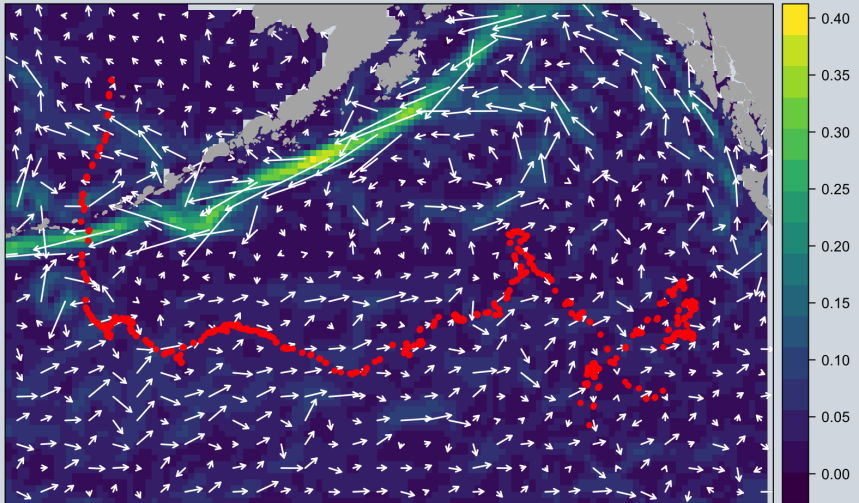
Spatial habitat for NFS migration

Mean Surface Wind



Spatial habitat for NFS migration

Mean Ocean Current



Multiple imputation and ctmcmove

Need Multiple Imputation because we don't directly observe $\mathcal{P} = f(\mu)$!

Multiple imputation and ctmcmove

Need Multiple Imputation because we don't directly observe $\mathcal{P} = f(\mu)$!

MI for CTMC models:

- 1 Simulate track from $[\mu|\mathbf{y}]$ (crawl funcs.)
- 2 Convert μ to \mathcal{P} (ctmcmove func.)
- 3 Form 'model data' \mathbf{z} , x_{ij} , m_{ij} , etc... (ctmcmove func.)
- 4 Fit CTMC model with `glm(...)` or `mgcv::gam(...)`
- 5 Save model object
- 6 repeat as necessary... (say 20-30 times)

Multiple imputation and ctmcmove

Need Multiple Imputation because we don't directly observe $\mathcal{P} = f(\mu)$!

MI for CTMC models:

- ① Simulate track from $[\mu|\mathbf{y}]$ (crawl funcs.)
- ② Convert μ to \mathcal{P} (ctmcmove func.)
- ③ Form 'model data' \mathbf{z} , x_{ij} , m_{ij} , etc... (ctmcmove func.)
- ④ Fit CTMC model with `glm(...)` or `mgcv::gam(...)`
- ⑤ Save model object
- ⑥ repeat as necessary... (say 20-30 times)

Multiple imputation and `ctmcmove`

Need Multiple Imputation because we don't directly observe $\mathcal{P} = f(\mu)$!

MI for CTMC models:

- 1 Simulate track from $[\mu|\mathbf{y}]$ (`crawl` funcs.)
- 2 Convert μ to \mathcal{P} (`ctmcmove` func.)
- 3 Form 'model data' \mathbf{z} , x_{ij} , m_{ij} , etc... (`ctmcmove` func.)
- 4 Fit CTMC model with `glm(...)` or `mgcv::gam(...)`
- 5 Save model object
- 6 repeat as necessary... (say 20-30 times)

Multiple imputation and `ctmcmove`

Need Multiple Imputation because we don't directly observe $\mathcal{P} = f(\mu)$!

MI for CTMC models:

- 1 Simulate track from $[\mu|\mathbf{y}]$ (`crawl` funcs.)
- 2 Convert μ to \mathcal{P} (`ctmcmove` func.)
- 3 Form 'model data' \mathbf{z} , x_{ij} , m_{ij} , etc... (`ctmcmove` func.)
- 4 Fit CTMC model with `glm(...)` or `mgcv::gam(...)`
- 5 Save model object
- 6 repeat as necessary... (say 20-30 times)

Multiple imputation and `ctmcmove`

Need Multiple Imputation because we don't directly observe $\mathcal{P} = f(\mu)$!

MI for CTMC models:

- 1 Simulate track from $[\mu|\mathbf{y}]$ (`crawl` funcs.)
- 2 Convert μ to \mathcal{P} (`ctmcmove` func.)
- 3 Form 'model data' \mathbf{z} , x_{ij} , m_{ij} , etc... (`ctmcmove` func.)
- 4 Fit CTMC model with `glm(...)` or `mgcv::gam(...)`
- 5 Save model object
- 6 repeat as necessary... (say 20-30 times)

Multiple imputation and ctmcmove

Need Multiple Imputation because we don't directly observe $\mathcal{P} = f(\mu)$!

MI for CTMC models:

- 1 Simulate track from $[\mu|\mathbf{y}]$ (crawl funcs.)
- 2 Convert μ to \mathcal{P} (ctmcmove func.)
- 3 Form 'model data' \mathbf{z} , x_{ij} , m_{ij} , etc... (ctmcmove func.)
- 4 Fit CTMC model with `glm(...)` or `mgcv::gam(...)`
- 5 Save model object
- 6 repeat as necessary... (say 20-30 times)

Multiple imputation and ctmcmove

Need Multiple Imputation because we don't directly observe $\mathcal{P} = f(\mu)$!

MI for CTMC models:

- 1 Simulate track from $[\mu|\mathbf{y}]$ (crawl funcs.)
- 2 Convert μ to \mathcal{P} (ctmcmove func.)
- 3 Form 'model data' \mathbf{z} , x_{ij} , m_{ij} , etc... (ctmcmove func.)
- 4 Fit CTMC model with `glm(...)` or `mgcv::gam(...)`
- 5 Save model object
- 6 repeat as necessary... (say 20-30 times)

Fitting the model

First, let's get an "imputer"

```
fit <- crwMLE(...)  
post_simulator <- crwSimulator(fit, predTime="10 mins", parIS=0)
```


Fitting the model

First, let's get an "imputer"

```
fit <- crwMLE(...)  
post_simulator <- crwSimulator(fit, predTime="10 mins", parIS=0)
```

From this we can make draws from $[\mu|y]$ (at exactly 10 min intervals)

```
ctcrw_path <- crwPostIS(post_simulator, fullPost=F)  
  
path <- list(  
  t=ctcrw_path$TimeNum[ctcrw_path$locType=="p"],  
  xy=as.matrix(  
    ctcrcw_path$alpha.sim[ctcrw_path$locType=="p",c("mu.x","mu.y")]  
  )  
)
```

Fitting the model

```
# Discretize the space
```

```
ctmc <- ctmcmove::path2ctmc(  
  xy=path$xy,t=path$t,  
  rast=examplerast,  
  zero.idx = zero.idx  
)
```

```
# Create model data for estimation
```

```
glm_data <- ctmc2glm_alt(ctmc,raster.list=raster.list,  
  zero.idx = zero.idx)
```

Fitting the model

```
form <- "z ~ crw + s(t,by=sst) +  
        s(t,by=wind_grad) + s(t,by=geo_curr_grad)"  
  
fit = mgcv::gam(  
  as.formula(form),  
  family="poisson", offset=log(tau), data=glm_data  
)
```

Fitting in parallel!

```
fit_ctmc <- function(ind=1, model, post.simulator,  
                      raster.list, zero.idx){  
  exemplarast <- raster.list[[1]][[1]]  
  ctcrw_path <- crwPostIS(...)  
  path <- list(t=..., xy=...)  
  ctmc <- ctmcmove::path2ctmc(...)  
  glm_data <- ctmc2glm_alt(...)  
  fit <- gam(model, family="poisson",  
             offset=log(tau), data=glm_data)  
  ...  
}
```

Fitting in parallel!

Execute over imputations in parallel

```
require(future.apply)
plan('multisession', workers=4)
mult.imp.fits <-
  future_lapply(1:25,
                FUN=fit_ctmc,
                model = model,
                post.simulator=post.simulator,
                raster.list=raster.list,
                zero.idx=zero.idx)
```

returns a list with 25 fitted model objects

Extracting the info from all the MI reps

- 1 Draw a posterior sample using MLE/MAP large sample $N(\hat{\beta}_r, \hat{\Sigma}_r)$,
 $r = 1, \dots, \text{num_rep}$
- 2 extract/manipulate sample as desired (e.g., effects predictions)
- 3 Concatenate samples together for full inference

Extracting the info from all the MI reps

- ① Draw a posterior sample using MLE/MAP large sample $N(\hat{\beta}_r, \hat{\Sigma}_r)$,
 $r = 1, \dots, \text{num_rep}$
- ② extract/manipulate sample as desired (e.g., effects predictions)
- ③ Concatenate samples together for full inference

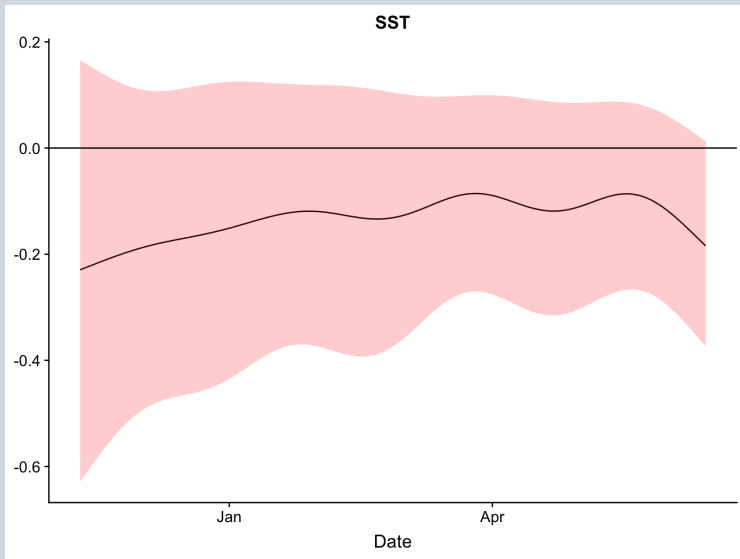
Extracting the info from all the MI reps

- 1 Draw a posterior sample using MLE/MAP large sample $N(\hat{\beta}_r, \hat{\Sigma}_r)$,
 $r = 1, \dots, \text{num_rep}$
- 2 extract/manipulate sample as desired (e.g., effects predictions)
- 3 Concatenate samples together for full inference

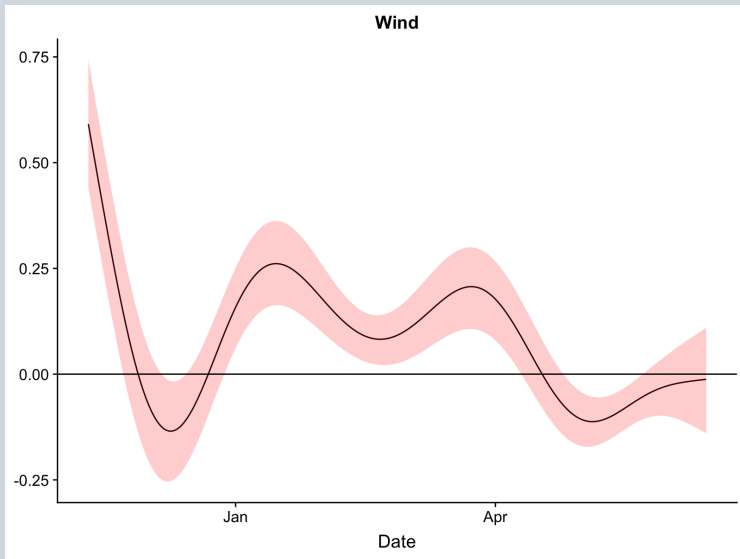
Extracting the info from all the MI reps

- 1 Draw a posterior sample using MLE/MAP large sample $N(\hat{\beta}_r, \hat{\Sigma}_r)$,
 $r = 1, \dots, \text{num_rep}$
- 2 extract/manipulate sample as desired (e.g., effects predictions)
- 3 Concatenate samples together for full inference

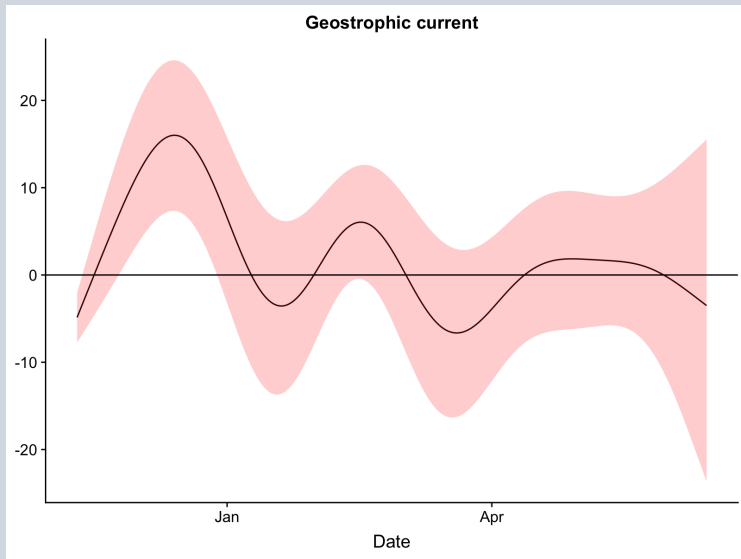
Habitat effects on cell transition



Habitat effects on cell transition



Habitat effects on cell transition



A little more math...

Rate matrix = $\mathbf{Q} = \{\lambda_{ij}\}$

Then we can obtain the transition matrix

$$\mathbf{P}(t) = [G(t)|G(0) = i] = \{\text{mexp}(\mathbf{Q}t)\}_i$$

Moreover, (for λ_{ij} constant over time, plus some other mathy requirements),

$$\mathbf{P}(\infty) = \mathbf{u} \propto \{A_i/\Lambda_i\},$$

where A_i is the limiting distribution of the embedded chain (Markov Chain of just the cell transitions)

Why \mathbf{u} ? It's the long-run probability that the animal will be in any particular cell the area. Sounds like a **utilization distribution**¹

¹Wilson et al. (2018). Estimating animal utilization densities using continuous-time Markov chain models. *Method Ecol and Evol* 9:1232-1240.

A little more math...

Rate matrix = $\mathbf{Q} = \{\lambda_{ij}\}$

Then we can obtain the transition matrix

$$\mathbf{P}(t) = [G(t)|G(0) = i] = \{\text{mexp}(\mathbf{Q}t)\}_i$$

Moreover, (for λ_{ij} constant over time, plus some other mathy requirements),

$$\mathbf{P}(\infty) = \mathbf{u} \propto \{A_i/\Lambda_i\},$$

where A_i is the limiting distribution of the embedded chain (Markov Chain of just the cell transitions)

Why \mathbf{u} ? It's the long-run probability that the animal will be in any particular cell the area. Sounds like a **utilization distribution**¹

¹Wilson et al. (2018). Estimating animal utilization densities using continuous-time Markov chain models. *Method Ecol and Evol* 9:1232-1240.

A little more math...

Rate matrix = $\mathbf{Q} = \{\lambda_{ij}\}$

Then we can obtain the transition matrix

$$\mathbf{P}(t) = [G(t)|G(0) = i] = \{\text{mexp}(\mathbf{Q}t)\}_i$$

Moreover, (for λ_{ij} constant over time, plus some other mathy requirements),

$$\mathbf{P}(\infty) = \mathbf{u} \propto \{A_i/\Lambda_i\},$$

where A_i is the limiting distribution of the embedded chain (Markov Chain of just the cell transitions)

Why \mathbf{u} ? It's the long-run probability that the animal will be in any particular cell the area. Sounds like a **utilization distribution**¹

¹Wilson et al. (2018). Estimating animal utilization densities using continuous-time Markov chain models. *Method Ecol and Evol* 9:1232-1240.

A little more math...

Rate matrix = $\mathbf{Q} = \{\lambda_{ij}\}$

Then we can obtain the transition matrix

$$\mathbf{P}(t) = [G(t)|G(0) = i] = \{\text{mexp}(\mathbf{Q}t)\}_i$$

Moreover, (for λ_{ij} constant over time, plus some other mathy requirements),

$$\mathbf{P}(\infty) = \mathbf{u} \propto \{A_i/\Lambda_i\},$$

where A_i is the limiting distribution of the embedded chain (Markov Chain of just the cell transitions)

Why \mathbf{u} ? It's the long-run probability that the animal will be in any particular cell the area. Sounds like a **utilization distribution**¹

¹Wilson et al. (2018). Estimating animal utilization densities using continuous-time Markov chain models. *Method Ecol and Evol* 9:1232-1240.

Where to from here?

- Residence times not exponentially distributed?
- Spatial-temporal covariates?
- Model selection?

Where to from here?

- Residence times not exponentially distributed?
- Spatial-temporal covariates?
- Model selection?

See:

Johnson, D. S., Pelland, N. & Sterling, J. T. (2021) A Continuous-time semi-Markov model for animal movement in a dynamic environment. *Ann. Appl. Stat.* 15:797–812

Additional reading and references

- Hooten, M. B., Johnson, D. S., Hanks, E. M., & Lowry, J. H. (2010). Agent-based inference for animal movement and selection. *JABES*, 15(4), 523-538.
- Johnson, D. S., Hooten, M. B., & Kuhn, C. E. (2013). Estimating animal resource selection from telemetry data using point process models. *J. of Anim. Ecol.*, 82(6), 1155-1164.
- Hanks, E. M., Hooten, M. B., & Alldredge, M. W. (2015). Continuous-time discrete-space models for animal movement. *The Ann. of Appl. Stat.*, 9:145-165.
- Hanks, E. M., & Hughes, D. A. (2016). Flexible discrete space models of animal movement. *arXiv:1606.07986*.
- Hooten, M. B., Johnson, D. S., McClintock, B. T., & Morales, J. M. (2017). *Animal movement: statistical models for telemetry data*. CRC Press. (Section 7.4)
- Wilson, K., Hanks, E., & Johnson, D. (2018). Estimating animal utilization densities using continuous-time Markov chain models. *Methods in Ecology and Evolution*, 9(5), 1232-1240.
- Johnson, D. S., Pelland, N. & Sterling, J. T. (2021) A Continuous-time semi-Markov model for animal movement in a dynamic environment. *Ann. Appl. Stat.* 15:797–812