Hidden Markov Models in Marine Sciences

fitting an HMM, state decoding, model checking

Vianey Leos Barajas & Marco Gallegos Herrada

Fitting an HMM to Marine Movement Data

Fitting an HMM to Marine Movement Data

- Now that we've learned about HMMs, we can talk about how to utilize HMMs when working with marine animal movement data.

- The most difficult step is to identify an HMM that **fits the data at hand**, i.e. is a good representation of the real movement process.

- We'll need to identify the key aspects of the data we are trying to capture. For example, an (unsupervised) HMM can be used to (at least) capture two aspects:
  (i) the marginal distribution of the data and (ii) the autocorrelation.

Fitting an HMM to Marine Movement Data

- Tailoring an HMM to data involves the following steps:

1. formulate candidate models, in particular specifying the class of state-dependent distributions and selecting the number of states, N

2. estimation of the model parameters

3. selecting between candidate models

4. checking the chosen model

Fitting an HMM

Inference for HMMs

- Once we select the number of states and forms of the state-dependent distributions, we aim to fit an HMM to data to recover two key components:

  1. **The parameters:** entries of the transition probability matrix, initial state distribution, estimates for the state-dependent distributions

  2. **Values of the states:** inferring the states that generated the observed data

How to fit the HMM

- There are broadly two ways of fitting an HMM — either estimating the parameters and states jointly, or separately. We'll do the latter.

- Using the likelihood, $\mathscr{L}(\boldsymbol{\theta}\,|\,\boldsymbol{y})$, i.e. the function of the parameters given the observations alone, we use maximum likelihood estimation to find the unique values $\boldsymbol{\theta}^*$ that maximize our function. Although in practice, we work on maximizing the log-likelihood, $\ell(\boldsymbol{\theta}\,|\,\boldsymbol{y}) = log\left(\mathscr{L}(\boldsymbol{\theta}\,|\,\boldsymbol{y})\right)$.

- In momentuHMM, the algorithm used to find the maximum likelihood estimate is the Newton-Raphson method.

Parameter constraints

- Contraints: transition probability matrix

  $1 > \gamma_{ij} > 0$ and $\sum_{i=1}^{m} \gamma_{ij} = 1$

- Constraints: state-dependent parameters

  Depend on distribution, but for example, a normal distribution has mean, $\mu \in \mathbb{R}$ and $\sigma > 0$

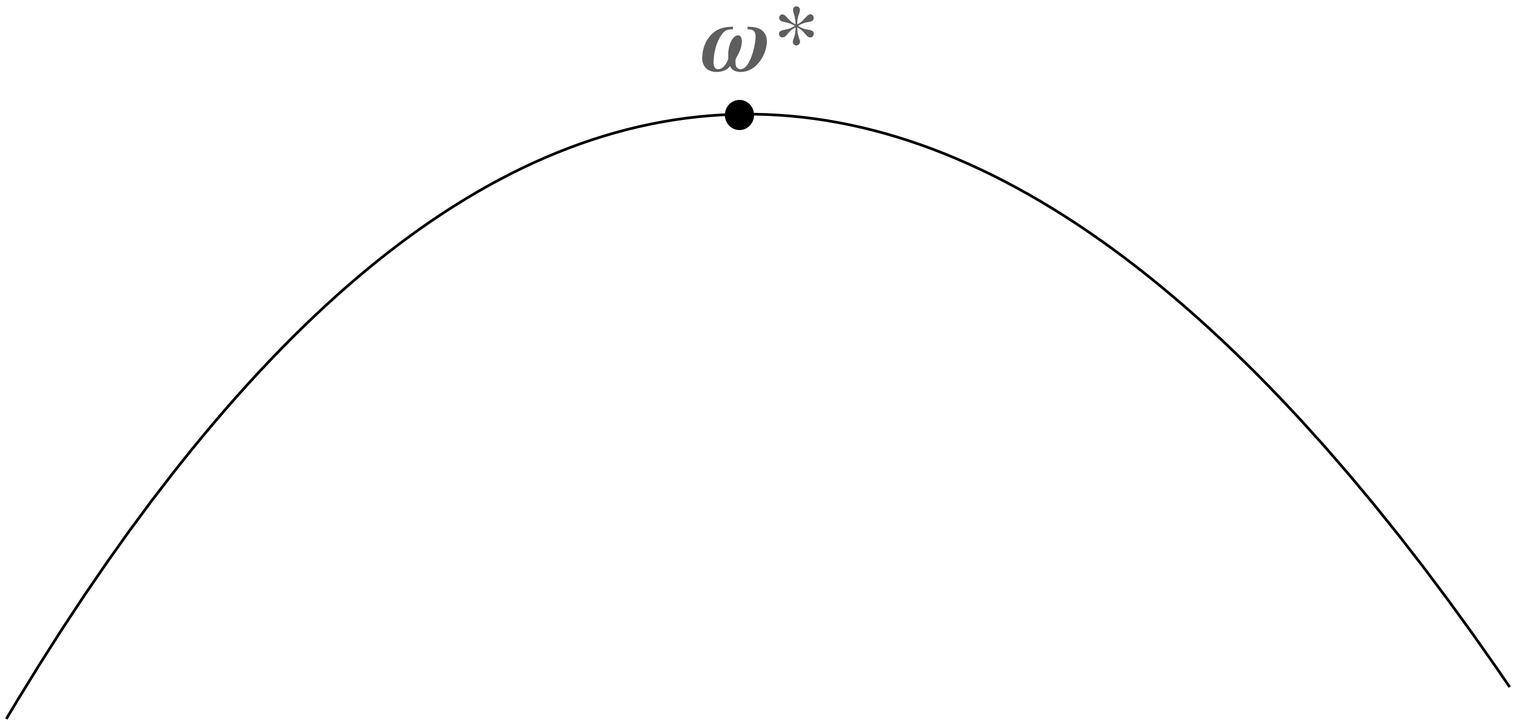
Natural Parameters to Working Parameters

- Finding the maximum likelihood values requires maximizing the log-likelihood.
- With constraints, our algorithm has to be wary of navigating into spaces of non-plausible values, e.g. $\sigma = -1$ is not defined

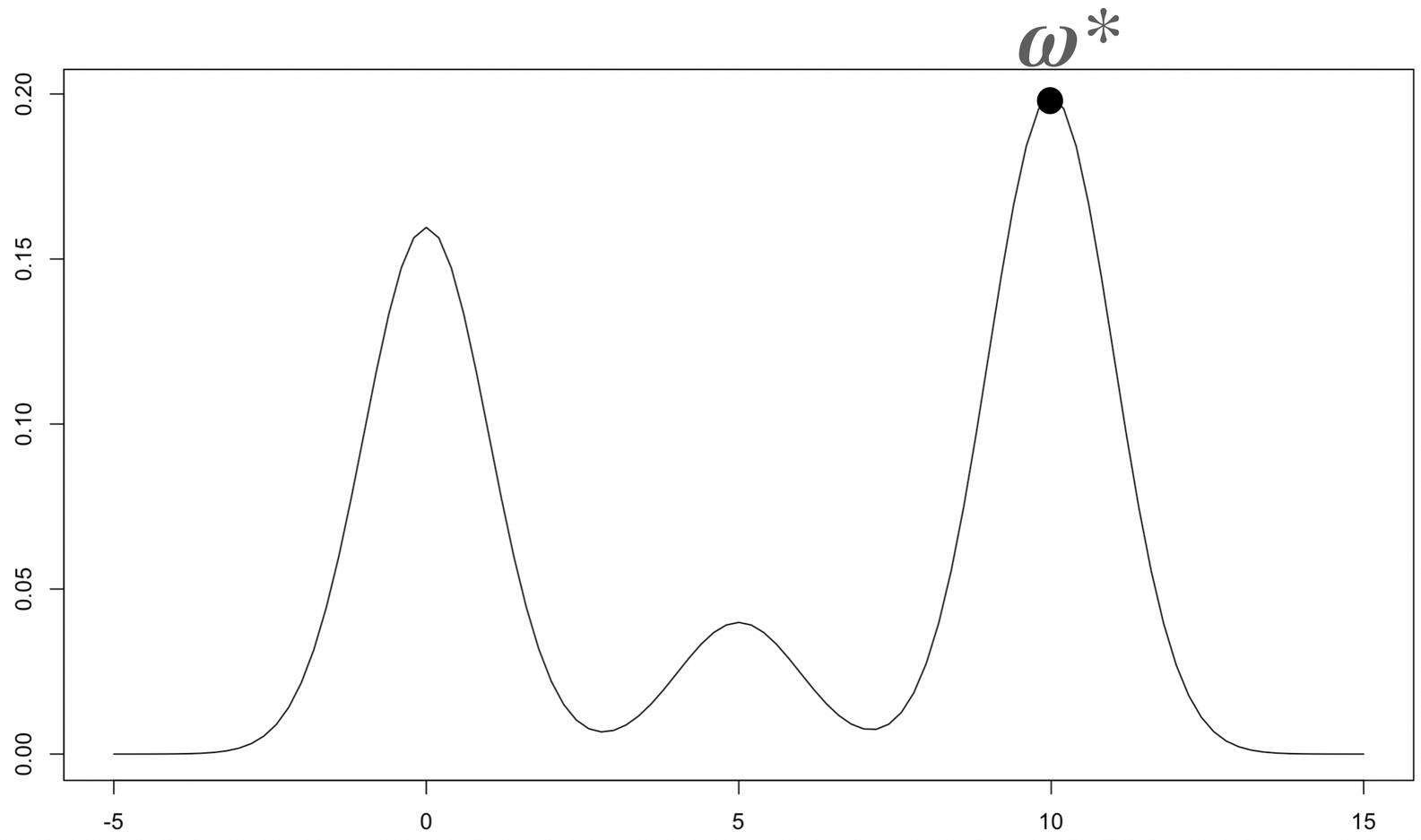
- **Natural parameters:** vector of parameters, $\boldsymbol{\theta}$, that we define in our model
- **Working Parameters:** vector of parameters, $\boldsymbol{\omega} = f(\boldsymbol{\theta})$, where $\omega_k \in \mathbb{R}$.

In practice, we find values that maximize $\ell(\boldsymbol{\omega} \,|\, \boldsymbol{y})$ and then compute $f^{-1}(\boldsymbol{\omega})^* = \boldsymbol{\theta}^*$

Suppose you're climbing a hill…



Local Maxima vs Global Maxima



Fitting an HMM In Practice

- The Newton-Raphson algorithm can do a good job of finding ‘local maxima’, but requires a starting point — we must provide this

- In momentuhmm, we must provide starting values for the state-dependent parameters (it will take care of starting values for the transition probability matrix when assumed time-homogenous)

- How do we know we found the ‘global maxima’?

  - Use many, many different starting values
  - Save the model output — we’ll need this!
  - Compare log-likelihood values — which is the maximum? Have you arrived there multiple times?

Uncertainty Quantification

- momentuhmm assumes asymptotic normality of the maximum likelihood estimator $\boldsymbol{\omega}^* \sim N(\boldsymbol{\omega}, \boldsymbol{\Sigma})$

- We can easily obtain confidence intervals for $\boldsymbol{\theta}^*$ from $\boldsymbol{\omega}^*$ using momentuhmm which implements the delta method (a first order approximation)

- Caveats: you may get warnings if the MLE is close to the boundary of the parameter space (e.g. when entries of the t.p.m. are estimated to be close to 0 or 1) In these cases, uncertainty quantification can be done other ways.

Missing Data

- Suppose some observations are missing OR we were not able to obtain a sample

$$\{y_1, y_2, \ldots, y_k, y_{k+2}, \ldots, y_T\}$$

- An HMM can easily accommodate that if we assume the data is ‘missing completely at random’

$$\boldsymbol{\delta}^T \mathrm{P}(y_1)\boldsymbol{\Gamma}\mathrm{P}(y_2)\ldots\boldsymbol{\Gamma}\mathrm{P}(y_k)\boldsymbol{\Gamma}\boldsymbol{\Gamma}\mathrm{P}(y_{k+2})\ldots\boldsymbol{\Gamma}\mathrm{P}(y_T)\mathbf{1}$$

Missing Data

- How much missing data is too much? How long should gaps be?

$$\{y_1, y_2, \ldots, y_k, y_{k+2}, \ldots, y_T\}$$

$$\{y_1, y_2, \ldots, y_k, y_{k+10}, \ldots, y_T\}$$

$$\{y_1, y_2, \ldots, y_k, y_{k+50}, \ldots, y_T\}$$

State Decoding

State Decoding

- State decoding is the process that we use to predict the states that generated our observations once we obtain ω^*
- There are two methods to do state decoding implemented in momentuhmm:

Global Decoding — Viterbi Algorithm**:** finds the optimal state sequence that could have generated the data

Local Decoding — Forward-Backward Algorithm**:** computes state probabilities at each time *t*

Global State Decoding: Viterbi Algorithm

- Consider Pr($C_1 = c_1, \ldots, C_T = c_T | y_1, \ldots, y_T$), and now assume we care to maximize this function

- We can try to find the values ($c_1^*, \ldots, c_T^*$) that maximize the probability, and this would give us the most likely sequence of states that generated our data!

- Doing this via brute force is impractical so we use the so-called Viterbi algorithm, a recursive algorithm that obtains ($c_1^*, \ldots, c_T^*$) this easily and quickly.

Local State Decoding: Forward-Backward Algorithm

- Consider Pr($C_t = c_t \,|\, y_1, \ldots, y_T$), so that we care to find the value of the state $c_t \in \{1,\ldots,m\}$ that maximizes this function

- The forward-backward algorithm also provides probabilities of states! We derive Pr($C_t = c_t \,|\, y_1, \ldots, y_T$) for each $c_t \in \{1,\ldots,m\}$ so we can have a better understanding of the “strength" of our state assignments.

- The forward-backward algorithm gets its name from the use of the forward and backward variables to compute these marginal state probabilities.

Model Checking

(Pseudo-) Residuals

- The concept of residuals can also be used to assess model fit for HMMs.
- However, these are not the residuals we know from linear regression.
- We begin with:

  - X_t has cumulative dist. function F_{X_t}, then $F_{X_t} \sim Uniform(0,1)$
  - $\phi^{-1}\left(F_{X_t}(X_t)\right) \sim N(0,1)$ if F_{X_t} is correct
  - $F_{X_t}(x_t) = Pr(X_t \leq x_t \,|\, X_1 = x_1, \ldots, X_t = x_t)$

- The **pseudo-residuals** are the quantities $\phi^{-1}\left(F_{X_t}(x_t)\right)$

Plotting (Pseudo-) Residuals

- Assumption of Normality:

  - assessed via Q-Q plots, histograms that assess assumption of normality, can also formally check for deviation from normality

- Autocorrelation:

  - residuals should not show any temporal dependence if the HMM has captured the autocorrelation in the data
  - we will assess the pseudo-residuals via ACF plots

Assessing State Predictions

- Unless we have part of our time series labelled, we can not assess our HMM in any way by the state predictions obtained.

- If our model checking checks out — great!

- If our states can serve as proxies for general behavior — great!!

- BUT, do keep in mind that the states are a wonderful byproduct of fitting an HMM to marine animal movement data. Unless we have labels, we can not assess if the model is actually useful for predicting specific behaviors of interest.

Next Lecture

- Model selection/comparison — selecting the number of states

- Including covariates

- Other useful extensions

  - random effects
  - including labeled data
  - hierarchical structures