Modeling Motor-Cargo Complexes Through Particle Filtering and EM Algorithm

Lauren Crow

November 20, 2017
Outline

Background

Mathematical Model

EM Algorithm

Particle Filtering

Implementation on Real Data

Current Goals

References
Background

- Motor proteins attached to a cargo move stochastically along a microtubule
- Motors can detach and attach from the microtubule
- Direction and speed depends on type of motor

Figure 1: Movement of two motor proteins attached to a cargo (Hancock, 2014)
Areas of Interest

- Prediction of regime, $J_t = \begin{cases} 1 & \text{attached/bound at each time, } t. \\ 2 & \text{unattached/unbound} \end{cases}$
- Locating binding sites, $Z(\tau(t))$, at each time, $t$, for a particular motor.
- Estimate model parameters in vector $\theta$.

Figure 2: Possible states of the particle (Feng, 2017)
The observed position of the particle is written as:

\[ Y_k = Z_k + \eta e_k, \quad k = 1, 2, 3, \ldots, \]

where

- \( Z_k \) is the true position of the cargo
- \( e_k \sim N(0, 1) \), and
- \( \eta \) is standard deviation of observational error.
Continuous-Time Model

The position of the particle is written as:

\[ Z(t) = Z(\tau(t)) \]

\[ + \left[ -\kappa \int_{\tau(t)}^{t} [Z(u) - Z(\tau(t))] \, du + \sqrt{2D_1} \int_{\tau(t)}^{t} dW(u) \right] I_{\{J(t)=1\}} \]

\[ + \left[ \int_{\tau(t)}^{t} \mu \, du + \sqrt{2D_2} \int_{\tau(t)}^{t} dW(u) \right] I_{\{J(t)=2\}}, \]

where

- \( \kappa \): linear-spring constant,
- \( D_i \): diffusion coefficient in regime \( i \), and
- \( \mu \): unbound drift term
Using the assumption that all regime switches occur at times of observations, we can approximate this as:

\[
Z_k = [Z_{\tau_k} + \phi(Z_{k-1} - Z_{\tau_k}) + \sigma_1 v_k] I\{J_k=1\}
+ [Z_{k-1} + \nu + \sigma_2 v_k] I\{J_k=2\},
\]

where, if \( \Delta \) is the time between observations,

- \( Z_k = Z(\Delta k) \),
- \( v_k \) are independent standard normal random variables,
- \( \phi = e^{\kappa \Delta} \),
- \( \sigma_1^2 = \frac{D_1}{\kappa} (1 - e^{-2\kappa \Delta}) \),
- \( \sigma_2^2 = 2D_2 \Delta \), and
- \( \nu = \mu \Delta \).
Properties of the Model

- Hidden Markov model
  - Observed: $Y_k$
  - Unobserved: $X_k = \{J_k, Z_k, Z_{\tau_k}\}$
- Position of particle is switching between ‘stuck’ and a directed random walk along microtubule
- Gaussian observational error
- Assumption that all regime switches occur at times of observation
- Can use Monte Carlo for simulation
EM allows us to estimate parameters through computation of maximum likelihood estimators.

Parameters of interest:

\[ \gamma_{ij} = P(J_k = j | J_{k-1} = i) = \frac{\lambda_i}{\lambda_i + \lambda_j} [1 - e^{-\Delta(\lambda_i + \lambda_j)}], \quad i = 1, 2 \]

\[ \phi = e^{-\kappa \Delta} \]

\[ \nu = \mu \Delta \]

\[ \sigma_1^2 = \frac{D_1}{\kappa} (1 - e^{-2\kappa \Delta}) \]

\[ \sigma_2^2 = 2D_2 \Delta \]

\[ \eta^2: \text{noise variance} \]

\[ \Rightarrow \theta = \{ \gamma_{12}, \gamma_{21}, \phi, \nu, \sigma_1^2, \sigma_2^2, \eta \} \]
EM Algorithm

- Let $\theta^{(h)}$ be the estimation of $\theta$ at iteration $h$.
- Repeat 1. and 2. until convergence:

1. **Expectation:**

   $$Q(\theta|\theta^{(h)}) = E_{\theta^{(h)}}[\log(L_X, Y(\theta))| Y_{0:N}]$$
   $$= \int \log(L_X, Y(\theta)) p_{\theta^{(h)}}(x_{0:N}| Y_{0:N}) dx_{0:N}$$

2. **Maximization:**

   $$\theta^{(h+1)} = \arg_{\theta} \max Q(\theta|\theta^{(h)})$$
EM Algorithm

- This results in the same estimates from the mle calculation but now each component is multiplied by the weight from the particle filter:

- **E Step:** \( Q_M(\theta | \theta^{(h)}) \)

\[
= \sum_{r=1}^{M} \sum_{k=1}^{N} \sum_{j=0}^{1} \log(\gamma_{ij}) I\{J_{k-1}\} w^r_N \\
+ \sum_{r=1}^{M} \sum_{k=1}^{N} \left[ -\log(\sigma_1^2) - \frac{\|Z_k - (Z_{\tau_k} + \phi(Z_{k-1} - Z_{\tau_k}))\|^2}{2\sigma_1^2} \right] I\{J_k=1\} w^r_N \\
+ \sum_{r=1}^{M} \sum_{k=1}^{N} \left[ -\log(\sigma_2^2) - \frac{\|Z_k - (Z_{k-1} + \nu)\|^2}{2\sigma^2} \right] I\{J_k=2\} w^r_N \\
+ \sum_{r=1}^{M} \sum_{k=0}^{N} \left[ -\log(\eta^2) - \frac{\|Y_k - Z_k\|^2}{2\eta^2} \right] w^r_N
\]
Particle Filtering

- Recall: \( X_k = \{ J_k, Z_k, Z_{\tau_k} \} \)

1. **Prediction Step:**

\[
p(x_k | Y_{0:k-1}) = \int p(x_k | x_{k-1}) p(x_{k-1} | Y_{0:k-1}) \, dx_{k-1}
\]

- Approximate \( p(x_{k-1} | Y_{0:k-1}) \) with particles \( \{ X_{k-1}^r \}_{r=1}^M \)
- Each particle \( X_{k-1}^r \) has an associated weight \( w_{k-1}^r \)
- Propagate particles forward according to \( X_k^r \sim p(x_k | X_{k-1}^r) \).

2. **Update Step:**

\[
p(x_k | Y_{0:k}) \propto p(Y_k | x_k) p(x_k | Y_{0:k-1})
\]
Figure 3: **Basic Particle Filter Example** - True location in black, predicted location in green, confidence bounds in red
EM Algorithm

- This results in the same estimates from the mle calculation but now each component is multiplied by the weight from the particle filter:

- **E Step**: $Q_M(\theta|\theta^{(h)})$

$$
= \sum_{r=1}^{M} \sum_{k=1}^{N} \sum_{j=0}^{1} \log(\gamma_{ij}) I_{\{J_{k-1}\}} w_N^r \\
+ \sum_{r=1}^{M} \sum_{k=1}^{N} \left[ - \log(\sigma_1^2) - \frac{\|Z_k - (Z_{\tau_k} + \phi(Z_{k-1} - Z_{\tau_k}))\|^2}{2\sigma_1^2} \right] I_{\{J_k=1\}} w_N^r \\
+ \sum_{r=1}^{M} \sum_{k=1}^{N} \left[ - \log(\sigma_2^2) - \frac{\|Z_k - (Z_{k-1} + \nu)\|^2}{2\sigma^2} \right] I_{\{J_k=2\}} w_N^r \\
+ \sum_{r=1}^{M} \sum_{k=0}^{N} \left[ - \log(\eta^2) - \frac{\|Y_k - Z_k\|^2}{2\eta^2} \right] w_N^r
$$

- **M-step**: Maximizes wrt each parameter of interest
Figure 4: Simulated Path and True vs. Predicted Regime
Implementation on Real Data: Kin2-Kin2
Figure 5: Convergence Plots from EM Algorithm:

\[ \nu \approx 2, \quad \sigma_1^2 \approx 0.7^2, \quad \sigma_2^2 \approx 16^2, \quad \gamma_{12} \approx 1 - 0.99, \]
\[ \gamma_{21} \approx 1 - 0.85, \quad \eta \approx 2.5, \quad \phi \approx 1 \]
Results - Regime Prediction

Figure 6: Predicted switches between bound and unbound
Current Goals

- Implement current method on different types of real data
  - Kin1-Kin1
  - Kin1-Kin2
  - Kin2-Kin2
  - Kin1-null
  - Kin2-null
  - Kin-Dyn

- Compare our method to the current one being implemented
References

Thank you
Particle Filter Algorithm

Let there be $M$ total particles and $N$ total time steps.

1. **Initialization**: The initial position is $Z^r_0 \sim N(0, \eta^2 I)$ for particle $r$. The initial regime for particle $r$ is $J^r_0 \sim \text{Bernoulli}(p)$, where $p$ comes from the transition matrix. Set $Z^r_{\tau_0} = Z^r_0$.

2. **Prediction**:
   
   - $Z^r_{\tau_k} = \begin{cases} 
   Z^r_{\tau_{k-1}} & \text{if } J^r_{k-1} = 1 \\
   Z^r_{k-1} & \text{if } J^r_{k-1} = 2 
   \end{cases}$
   
   - $J^r_k \sim p(j_k|J^r_{k-1})$
   
   - $Z^r_k \sim p(z_k|Z^r_{k-1}, Z^r_{\tau_k}, J^r_k)$.

3. **Update**: Compute weights $w^r_k = p(Y_k|Z^r_k)w^r_{k-1}$ and normalize.

4. **Resample**: Resample $M$ particles from the current set of particles using normalized weights as the probabilities of being selected.

(Bernstein, 2016).