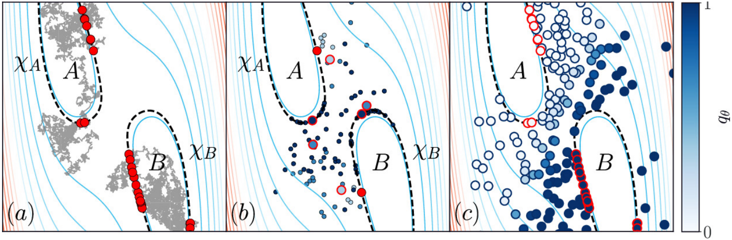
Learning Multi-Basin Committors using Neural Networks and Markov State Models

Based on [1] by Andrew R. Mitchell, Grant M. Rotskoff (Stanford)

Connor Blake, May 23, 2025

Motivation

- Committors are the "ideal reaction coordinate"
- Rare event sampling is hard, important
- ML has seen good results recently [2]



A and B basins [1]

Notation

$$x \in \Omega \subset \mathbb{R}^d$$

$$C_k \subset \Omega \quad C_k \cap C_{k'} = \emptyset \;\; \forall k, k' \in [K]$$

$$\Omega^C = \Omega \setminus (\cup_k C_k)$$

$$dx_t = -\nabla U(x_t)\, dt + \sqrt{2\beta^{-1}}\, dW_t$$

$$f : \Omega \to \mathbb{R}$$

$$\mathcal{L} f(x) = -\nabla U(x) \cdot \nabla f(x) + \beta^{-1}\, \nabla^2 f(x)$$

$$\frac{\partial f(x,t)}{\partial t} = \mathcal{L} f(x,t), \quad f(x,0) = f_0(x)$$

$$\mathcal{T}_\tau f(x_0) = \mathbb{E}\big[f(x_\tau^{x_0})\big]$$

Committor Properties

$$p_k(x) = P(\text{trajectory hits basin } k \text{ before any other basin})$$

$$q_k(x) = 1 - p_k(x)$$

$$\sum_k p_k(x) = 1 \quad \forall x \in \Omega$$

$$\left.\begin{aligned} \mathcal{L} p_k(x) &= 0 \quad x \in \Omega^C \\ p_k(x) &= 1 \quad x \in C_k \\ p_k(x) &= 0 \quad x \in \bigcup_{k' \neq k} C_{k'} \end{aligned}\right\} \text{FEM}$$

$$\begin{aligned} p_k(x_0) &= E[p_k(x_\tau^{x_0})] \quad x \in \Omega^C \\ &\approx \frac{1}{n} \sum_{i=1}^{n} p_k(x_\tau^{x_0,i}) \quad \text{(CLT)} \end{aligned}$$

Log Committor Algorithm, $A \to B$

Hill Relation: $r_{AB} = j_{A^-} \int_{\partial A} q_A(x) dx$

Their Goal: $q_{\theta,A} : \Omega \to \mathbb{R} \xrightarrow{\text{sigmoid}} [0,1]$

$$q_A(x) + q_B(x) = 1 \quad \{A, B\} = [K]$$

$$q_A(x_0) \approx \frac{1}{n} \sum_{i=1}^{n} q_A(x_\tau^{x_0,i})$$

$$\log(q_A(x_0)) \approx \log\left(\frac{1}{n} \sum_{i=1}^{n} q_A(x_\tau^{x_0,i})\right) \quad \text{(biased)}$$

Loss: $L_{nk\tau}(\theta) = \frac{1}{|[K]|} \sum_{k \in [K]} \frac{1}{2n} \sum_{i=1}^{n} \log\left(\frac{q_{\theta,k}(x_i)}{\frac{1}{M} \sum_{m=1}^{M} q_{\theta,k}(x_\tau^{x_i,m})}\right)^2$

Base Algorithm Results

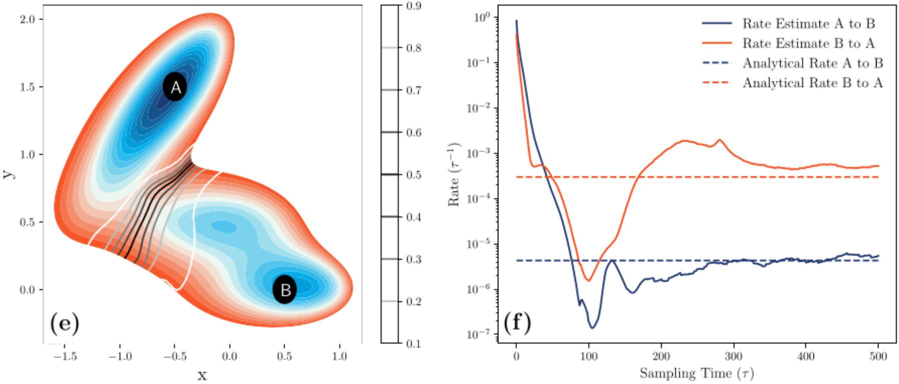


Figure: Muller-Brown Potential e) MLSE Committor f) Rate estimates [1]

Base Algorithm Results (cont.)

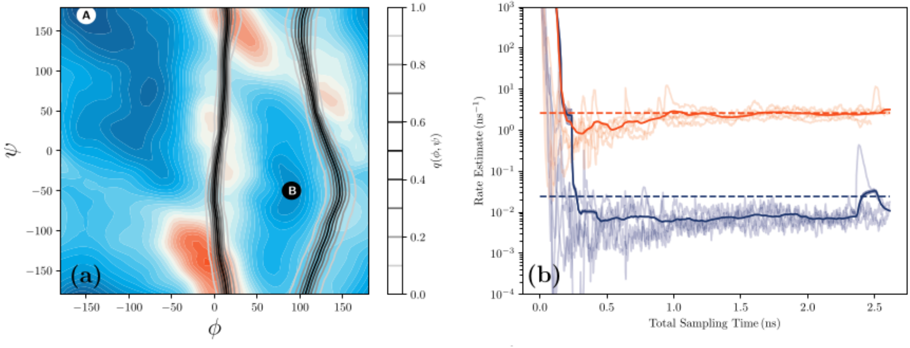


Figure: Dialanine peptide a) MLSE Committor on ϕ, ψ b) Rate estimates [1]

Base Algorithm Problems

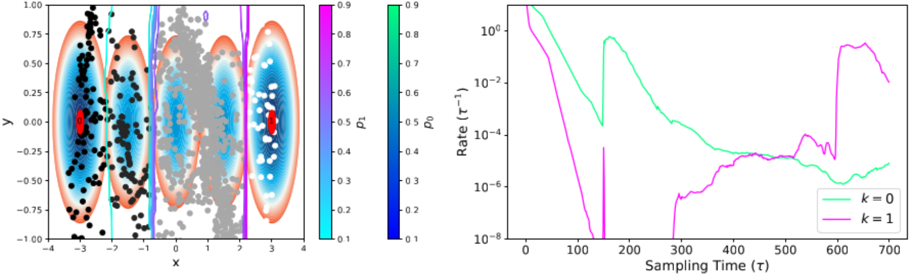


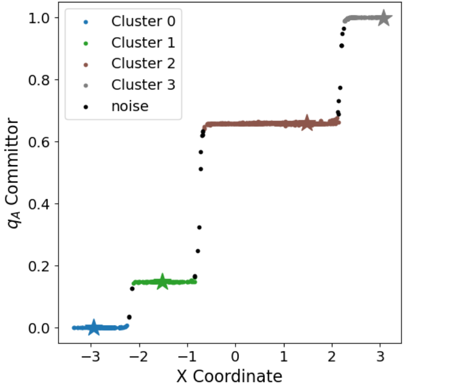
Figure: 5 Linear Wells a) MLSE Committor b) "Rate estimates". There is no observed stability in the committor-compute exit rates from the basins 0 and 1.

Fixing the Base Algorithm

1. Run a short sample from base algorithm
2. Run DBSCAN on q_A committor values ($\epsilon = .01$) into K clusters (including originals 2)
3. $$x_{k,c} = \underset{x_{k,i}}{\arg\min}\{U(x_{k,i})\}$$
4. Rerun simulation with K basins to sample from starting from $x_{k,c}$ to train

$$p_{\theta,k} : \Omega \to \mathbb{R}^K \xrightarrow{\text{softargmax}} [0,1]^K$$

Bad Rate Estimator, Good Metastability Estimator



DBSCAN on q_A committor. Note: the X coordinate here is for visualization purposes only. The clustering algorithm is run on purely the one-dimensional committor prediction vector of the stored points.

Improved Algorithm Results

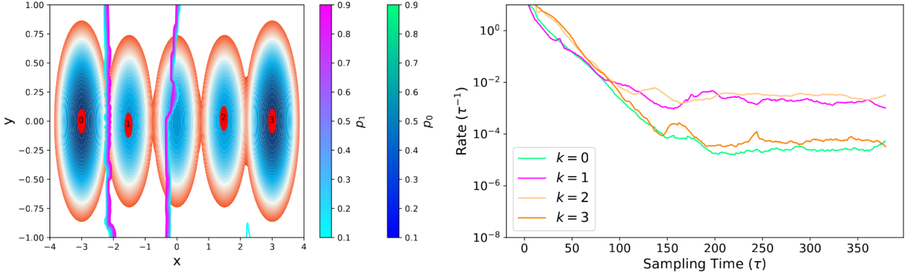


Figure: 5 Linear Wells a) MLSE Committor b) Rate estimates. The exit rates for each basin rapidly converge due to the improved committor training method.

Composite Rates from Continuous Time Markov Chains

$$Q_{ij} = k_{i\to j}, \quad Q_{ii} = -\sum_{j\neq i} k_{i\to j} = -j_{k^-}\int_{\partial C_k} q_k(x)dx \quad \text{(Hill Relation)}$$

$$\sum_k Q_{ik}\, m_{kj} = -1, \quad i \neq j; \quad m_{jj} = 0$$

$$T_{ij} \sim \mathrm{Exp}(k_{ij}) \implies \mathbb{E}[T_{ij}] = \frac{1}{k_{ij}} \implies k_{ij} = \frac{1}{m_{ij}}$$

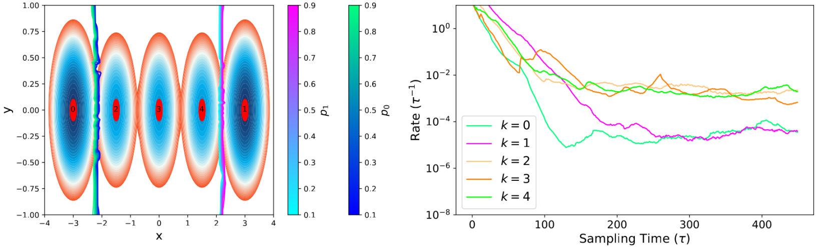


Figure: Rates from 0 to 1 can now be computed with a CTMC

References I

Andrew R. Mitchell and Grant M. Rotskoff.
Committor guided estimates of molecular transition rates.
arXiv preprint arXiv:2408.05879, 2024.

John Strahan, Chatipat Lorpaiboon, Jonathan Weare, and Aaron R. Dinner.
Bad-neus: Rapidly converging trajectory stratification, 2024.