Dimensionality reduction for MSM construction

BACKGROUND

We would like to model the long-timescale kinetics of biomacromolecules, since processes like ligand-binding and protein-folding occur on timescales much larger than the simulation timesteps. This kinetic behavior is typically dominated by "rare events." A successful modeling approach that exploits this structure is the construction of Markov State Models (MSMs) from atomistic simulations. Essentially, this involves decomposing the state space of the molecule into a discrete set of long-lived (metastable) states, and estimating the transition rates between these states. Estimating the transition rates basically reduces to counting (with some complications, as well as sophisticated Bayesian estimation methods). The main challenge is identifying a meaningful decomposition of the state space.

Decomposing a macromolecule's state space is a problem of metric learning and clustering. We want a distance function that accepts two protein conformations, and will produce a large distance if they are "kinetically distant" and a small number if they are "kinetically close"– i.e. if they can interconvert quickly. A naive distance metric (e.g. Euclidean distance between raw position vectors) may not be sensitive to long-timescale behavior, and may not be invariant to things like rotation and translation. Metrics like RMSD are conveniently invariant to rotation and translation, but may be sensitive to high-speed fluctuations.

Design considerations

- (+) **Bayesian** – To play well with other parts of the MSM pipeline, we would like to be able to perform model selection in a Bayesian framework, as well as to marginalize over nuisance parameters and propagate uncertainty to future steps.

- (+) **Computationally efficient** – We would like to be able to apply any proposed method to long trajectories with up to millions of simulation frames. The computational cost of a method should not grow much faster than $O(Nd^2)$ with N the number of frames and d the number of atoms.

- (+) **Nonlinear** – Effectively describing slow molecular motion using a small number of col-

lective variables may be impossible if these variables are constrained to be linear functions of the input coordinates.

- (+) **Rotation-invariant** – The distance between two protein conformations should be invariant to rigid-body rotations and translations.

- (-) **Require manual feature engineering** – We would like a method to require minimal imposition of

RESEARCH PLAN

We would like to evaluate a set of metric learning approaches for the purpose of building Markov State Models of biomolecular kinetics.

Measuring performance

We can measure performance of an MSM by examining the convergence of the implied timescales and by the Generalized Matrix Rayleigh Quotient (GMRQ). These measure the ability of an MSM to accurately model the long-timescale behavior of the protein

[JHF: Include a figure]

Generalized Matrix Rayleigh Quotient (GMRQ)

Input coordinates

- Raw Cartesian coordinates

- Cartestian coordinates after alignment onto a fixed reference structure

- Pairwise interatomic distances – Often restricted to a subset of the atoms in the molecule, esp. the heavy atoms.

-

Mahalanobis distances

In some sense the simplest distance functions we can consider are **Mahalanobis distances**

$$d_X(a, b) \equiv (a - b)^T X (a - b)$$

where X is a symmetric p.s.d. matrix.

A subset of these are **squared weighted Euclidean** distances, where X is diagonal.

Principal Component Analysis (PCA)

Find a matrix M with orthonormal columns (defining a projection function $f : X \mapsto M^T X$) that minimizes squared reconstruction error ($M = \text{argmin}_{M \in \mathcal{O}^{d \times r}} \|X - MM^T X\|_F^2$).

This can be solved in closed form by solving an eigenvalue problem: columns of M are the leading eigenvectors of the data covariance matrix.

Time-structured Independent Component Analysis (tICA)

tICA is a method for finding low-dimensional linear projections that best capture the slow processes at a given lag-time τ.

Given a mean-free time-series $\boldsymbol{y}(t)$, find a matrix $\mathbf{M}$ that solves the following generalized eigenvalue problem involving the data covariance matrix $\mathbf{C}(0)$ and the time-lagged covariance matrix $\mathbf{C}(\tau)$: [1]

$$\mathbf{C}(\tau)\mathbf{M} = \mathbf{C}(0)\mathbf{M}\mathbf{\Lambda}$$

We can interpret $\mathbf{M}$ as containing linear approximations to the eigenfunctions of the full MD propagator, motivating its use in constructing MSMs, and we can interpret the associated eigenfunctions as indicators of the relaxation timescales in those directions.

We can also interpret tICA as solving an optimization problem over matrices. Note that the columns of $\mathbf{M}$ are not necessarily orthogonal.

[JHF: Problems: when the lag-time τ is longer than the slowest relaxation time in the system, the some of the eigenvalues may be negative, corresponding to “flipping” processes.]

[1] $c_{ij}(\tau) = \langle y_i(t) y_j(t + \tau) \rangle_t$

Weighted RMSD / RMSD-similars

Given two sets of zero-centered 3D coordinates, we can find the optimal rigid-body rotation to align the two[2], then sum up the squared distance between every corresponding pair of atoms, i.e.

$$\text{RMSD}(X, Y) = \min_R \sum_i \|X_i R - Y_i\|^2$$

for rotation matrices R.

We can also weight the individual atoms to yield "weighted RMSD":

$$\text{wRMSD}(X, Y, w) = \min_R \sum_i w_i \|X_i R - Y_i\|^2$$

A function that computes something similar to RMSD is the so-called "Binet-Cauchy kernel:"

$$\text{BC}(X, Y) = \frac{\det(X^T Y)}{\sqrt{\det(X^T X)\det(Y^T Y)}}$$

.

Interpretation: this kernel is computing the cosine of the grassmann vectors of X and Y.

We can then parametrize these distances by multiplying in a weight matrix:

$$\text{wBC}(X, Y; W) = \frac{\det(X^T W Y)}{\sqrt{\det(X^T W X)\det(Y^T W Y)}}$$

. where W is symmetric, p.s.d.?

Large-margin kinetic metric learning

Assume we have a family of distance functions $d_w(\cdot, \cdot)$ parametrized by a weight vector w. We want to find the weight vector that maximizes the distance between kinetically distant points and minimizes the distance between kinetically close points.

Given an observed sequence of points $\{x_t\}$ and two lag-times τ_1 and τ_2 ($\tau_2 \gg \tau_1$), we can phrase this as an optimization problem in terms of a triplet loss function:

$$w^* = \text{argmin}_w \sum_t \ell(w|(x_t, x_{t+\tau_1}, x_{t+\tau_2}))$$

where, for example, we might take the loss function to be:

$$\ell(w|(x_t, x_{t+\tau_1}, x_{t+\tau_2})) \equiv d_w(x_t, x_{t+\tau_1}) - d_w(x_t, x_{t+\tau_2})$$

[2] using the Kabsch algorithm

Triplet network

We can train a deep feed-forward neural network to embed input vectors into a low-dimensional space that minimizes some loss function over triplets of time-lagged observations.

Generative embeddings?

We expect that the sequence of observed high-dimensional snapshots are projections of a Markov process evolving in a hidden (and potentially lower-dimensional) latent space.

A natural modeling approach is then to marginalize over the hidden coordinates to estimate the latent-to-observed projection function. If the projection function is invertible, then this now lets us

DYNAMICS MODELS

Markov State Models (MSMs)

We assume the data sequence is generated by

Projected and Hidden Markov Models (PMMs and HMMs)

Reduced-Rank Hidden Markov Models (RR-HMMs)