

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** – 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. We would also like to use uncertainty estimates to intelligently drive sampling to regions.
- (+) **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 collective 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
- Cartesian coordinates after alignment onto a fixed reference structure
- Pairwise distances – E.g. between residues or atoms. Often restricted to a subset of the atoms in the molecule, esp. the heavy atoms.
- Distribution of reciprocal interatomic distances
- Dihedral angles

Mahalanobis distances

Distance functions defined by affine transformations 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 ($M = \operatorname{argmin}_{M \in \mathcal{O}^{d \times r}} \|X - M M^T X\|_F^2$).

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 $y(t)$, find a matrix M that solves the following generalized eigenvalue problem involving the data covariance matrix $C(0)$ and the time-lagged covariance matrix $C(\tau)$:¹

$$C(\tau)M = C(0)M\Lambda$$

We can interpret 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 eigenvalues as indicators of the relaxation timescales in those directions.

We can also interpret tICA as solving an optimization problem over matrices.²

This generalized eigenvalue problem can be interpreted as an optimization problem. Given two $n \times n$ matrices (A, B) , a generalized eigenvalue problem is to find pairs (λ, v) that satisfy $Av = \lambda Bv$. Finding the dominant eigenspace corresponds to minimizing the generalized

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

² Note that the columns of M are not required to be orthogonal.

72 Rayleigh quotient:

$$f(Y) = \text{tr}(Y^T A Y (Y^T B Y)^{-1}) = \text{tr} \left(\frac{Y^T A Y}{Y^T B Y} \right) \quad (1)$$

73 where Y is a full-rank $n \times p$ matrix.

74 Y_* spans the leftmost invariant subspace³ of (A, B) if and only if Y_* is a global minimizer of
75 the generalized Rayleigh quotient 1. [JHF: Ref: Absil, Optimization on Matrix Manifolds]

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

So we can interpret tICA as minimizing the following objective function involving a trace of quotients:

$$f_{\text{tICA}}(M) = \text{tr} \left(\frac{M^T \mathbf{C}(\tau) M}{M^T \mathbf{C}(0) M} \right)$$

78 A more appropriate objective might be the following, involving a quotient of traces:

$$f_{\text{MAF}}(M) = \frac{\text{tr}(M^T \mathbf{C}(\tau) M)}{\text{tr}(M^T \mathbf{C}(0) M)}$$

79 For related problems, including maximum autocorrelation factors (MAF) and linear discrim-
80 inant analysis (LDA), Cunningham and Ghahramani (2015) compared objective functions based
81 on trace-of-quotients vs. quotient-of-traces. The quotient-of-traces objective cannot be formu-
82 lated as an eigenvalue problem.

83 [JHF: See also: Fukunaga, 1990,]

84 **Weighted RMSD / RMSD-similars**

85 *RMSD*

Given two collections of zero-centered 3D coordinates, $X, Y \in \mathbb{R}^{N \times 3}$, we can find the optimal rigid-body rotation to align the two⁴, then sum up the squared distance between every corresponding pair of atoms, i.e.

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

86 where the minimization is over 3×3 rotation matrices R .

³ i.e. Y_* spans the same space spanned by the dominant generalized eigenvectors

⁴ using the Kabsch algorithm

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

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

88 where $\mathbf{w} \in \mathbb{R}^N$.

89 This flexibility allows us to reduce the influence of

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

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

91 .

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

We can then parametrize these distances by multiplying in an $N \times N$ weight matrix W :

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

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

95 **Large-margin kinetic metric learning**

96 Assume we have a family of distance functions $d_{\boldsymbol{\theta}}(\cdot, \cdot)$ parametrized by a vector $\boldsymbol{\theta}$. We want
97 to find the weight vector that maximizes the distance between kinetically distant points and
98 minimizes the distance between kinetically close points.

Given an observed sequence of points $\{\mathbf{y}_t\}$ 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:

$$\boldsymbol{\theta}^* = \operatorname{argmin}_{\boldsymbol{\theta}} \sum_t \ell(\boldsymbol{\theta} | (\mathbf{y}_t, \mathbf{y}_{t+\tau_1}, \mathbf{y}_{t+\tau_2}))$$

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

$$\ell(\boldsymbol{\theta} | (\mathbf{y}_t, \mathbf{y}_{t+\tau_1}, \mathbf{y}_{t+\tau_2})) \equiv d_{\boldsymbol{\theta}}(\mathbf{y}_t, \mathbf{y}_{t+\tau_1}) - d_{\boldsymbol{\theta}}(\mathbf{y}_t, \mathbf{y}_{t+\tau_2})$$

Triplet network

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

In other words, we consider parametrized mappings of the form

$$f_{\theta} : \mathbf{y} \mapsto \sigma(W_n \dots \sigma(W_2 \sigma(W_1 \sigma(W_0 \mathbf{y} + \mathbf{b}_0) + \mathbf{b}_1) + \mathbf{b}_2) \dots + \mathbf{b}_n)$$

where the input is repeatedly multiplied by a weight matrix W_i and passed through a (nonlinear) activation function σ , i.e. θ contains $W_0, \dots, W_n, \mathbf{b}_0, \dots, \mathbf{b}_n$, $W_i \in \mathbb{R}^{s_i \times s_{i-1}}$ are real weight matrices, $\mathbf{b}_i \in \mathbb{R}^{s_i}$ are bias vectors, and σ_i are activation functions (e.g. sigmoid or tanh), applied elementwise. If all σ_i are linear, then the mappings are linear. The shape of each weight matrix W_i determines the “width” of the subsequent layer.

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 a Markov jump process over discrete clusters.

Projected and Hidden Markov Models (PMMs and HMMs)

Reduced-Rank Hidden Markov Models (RR-HMMs)