Dimensionality reduction for MSM construction

BACKGROUND

We would like to construct simple models of the long-timescale kinetics of biomacromolecules, since processes like ligand-binding and protein-folding occur on timescales much larger than those accessible by brute-force molecular dynamics. This kinetic behavior is typically dominated by “rare events,” in which the system spends most of its time in a free energy well and rarely transitions into another free energy well. A successful modeling approach that exploits this structure is the construction of Markov State Models (MSMs) from atomistic simulations. Essentially, this involves performing a series of long simulations with diverse initial conditions, then decomposing the state space of the molecule into a discrete set of long-lived (metastable) states, then 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 kinetically 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. Naive geometric distance metrics (e.g. Euclidean distance between raw position vectors) may not be sensitive to degrees of freedom that decorrelate on long timescales, 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.

FULL EXAMPLE OF BUILDING AND SCORING AN MSM ON AN EXAMPLE DATASET

RESEARCH PLAN

We would like to evaluate a set of metric learning approaches for the purpose of building Markov State Models (MSMs) 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)

McGibbon and Pande, 2015 (`http://arxiv.org/pdf/1407.8083v3.pdf`) introduced an objective function for low-rank estimators of slow dynamics.

DESIGN CONSIDERATIONS

- **Rotation-invariant** – The distance between two protein conformations should be invariant to rigid-body rotations and translations.
- **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 other pipeline steps. Additionally, if we have access to well-calibrated uncertainty estimates, we could in principle use these estimates to perform active sampling.
- **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.
- **Physically interpretable** – We would like to be able to examine the learned metric, both to gain insight into the slow relaxation processes and to . For example, if the output coordinates are sparse functions of the input features.

- **Minimal feature engineering** – We would like a method to require minimal pre-specification of kinetically important "features" or collective variables.

MODELING STRATEGIES

The input objects are configurations, N-by-3 matrices containing the raw 3D coordinates of each of the N atoms in the molecule.

Input coordinates

- Raw Cartesian coordinates
- Cartestian 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

Linear transformations

We can consider distance functions defined by affine transformations,

$$d_M(a,b) \equiv \|M^Ta - M^Tb\|_2$$

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

Principal Component Analysis (PCA)

Find a matrix $M \in \mathbb{R}^{d\times r}$ with orthonormal columns (defining an orthogonal projection function $f : X \mapsto M^TX$) that minimizes squared reconstruction error ($M = \mathrm{argmin}_{M\in\mathcal{O}^{d\times r}} \|X - MM^TX\|_F^2$). [1]

[1] Equivalently, we can minimize $-\text{tr}(M^TXX^TM) \propto -\text{tr}(M^T\Sigma M)$, where Σ is the empirical covariance matrix. This can be solved in closed form by solving an eigenvalue problem: greedily select the dominant r eigenvectors in the eigenvalue decomposition $\Sigma = Q\Lambda Q^T$

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 M that solves the following generalized eigenvalue problem involving the data covariance matrix $\mathbf{C}(0)$ and the time-lagged covariance matrix $\mathbf{C}(\tau)$: [2]

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

M has been interpreted as containing linear approximations to the eigenfunctions of the full MD propagator, motivating its use in constructing MSMs. Each eigenvector corresponds to a direction in which the system relaxes slowly, and the associated eigenvalues are indicators of the timescales of relaxation in those directions.

We can also interpret tICA as solving an optimization problem over matrices. [3]

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 Rayleigh quotient:

$$f(Y) = \text{tr}(Y^T AY(Y^T BY)^{-1}) = \text{tr}\left(\frac{Y^T AY}{Y^T BY}\right) \tag{1}$$

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

Y_* spans the leftmost invariant subspace [4] of (A, B) if and only if Y_* is a global minimizer of the generalized Rayleigh quotient 1. [JHF: Ref: Absil, Optimization on Matrix Manifolds]

[JHF: Problems: when the lag-time τ is longer than the slowest relaxation time in the system, then 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:

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

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

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

[3] Note that the columns of M are not required to be orthogonal in the tICA problem.

[4] i.e. Y_* spans the same space spanned by the dominant generalized eigenvectors

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

For related problems, including maximum autocorrelation factors (MAF) and linear discriminant analysis (LDA), Cunningham and Ghahramani (2015) compared objective functions based on trace-of-quotients vs. quotient-of-traces. Often these objectives are mistakenly treated as equivalent, although the quotient-of-traces objective cannot be formulated as an eigenvalue problem.

Kernel tICA (ktICA)

Similar to the kernelized version of PCA, we can define a dual version of tICA, and solve it with arbitrary choice of kernel function.

[JHF: See also the references in Cunningham and Ghahramani: Fukunaga, 1990,]

Weighted RMSD / RMSD-similars

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 two[5], then sum up the squared distance between every corresponding pair of atoms, i.e.

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

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

Weighted RMSD

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

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

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

[5] using the Kabsch algorithm

This allows us to “de-emphasize” parts of the protein that may fluctuate rapidly, and to “emphasize” parts of the protein that indicate the long-timescale relaxation processes.

How should we choose $\boldsymbol{w}$?

Binet-Cauchy Kernel

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

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

.

Like RMSD, this function also invariant to rotation.

Interpretation: this kernel is computing the cosine between the grassmann vectors of X and Y. b We can then parametrize these distances by multiplying in an $N \times N$ weight matrix W:

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

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

Large-margin kinetic metric learning

Assume we have a family of distance functions $d_{\boldsymbol{\theta}}(\cdot, \cdot)$ parametrized by a vector $\boldsymbol{\theta}$. We want to find the weight vector that maximizes the distance between kinetically distant points and minimizes the distance between kinetically close points. In other words, we want the metric to “pull” similar points together and “push” dissimilar points far apart.

Given an observed sequence of points $\mathcal{D} = \{\boldsymbol{y}_t\}$, we can extract triplet labels of the form (a, b, c) where a is more similar to b than to c. One way to do this is to select two lag-times τ_1 and τ_2 ($\tau_2 \gg \tau_1$), and provie triplet labels $(\boldsymbol{y}_t, \boldsymbol{y}_{t+\tau_1}, \boldsymbol{y}_{t+\tau_2})$.

We can then phrase this as an optimization problem in terms of a sum over triplets of a loss function:

$$\boldsymbol{\theta}^* = \text{argmin}_{\boldsymbol{\theta}} \sum_t \ell(\boldsymbol{\theta} | (a, b, c)) \tag{2}$$

where, for example, we might take the loss function to be linear[6]:

$$\ell(\boldsymbol{\theta} | (a, b, c)) \equiv d_{\boldsymbol{\theta}}(a, b) - d_{\boldsymbol{\theta}}(a, c)$$

[6] This would be a bad choice in practice since we could always improve the value of the loss function by globally scaling the outputs.

or, as in Weinberger et al., 2005:

$$\ell(\boldsymbol{\theta}|(a,b,c)) \equiv d_{\boldsymbol{\theta}}(a,b)^2 + \lambda(1 + d_{\boldsymbol{\theta}}(a,b)^2 - d_{\boldsymbol{\theta}}(a,c)^2)$$

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

$$f_{\boldsymbol{\theta}} : \boldsymbol{y} \mapsto \sigma_n \left(W_n \dots \sigma_2 \left(W_2 \sigma_1 \left(W_1 \sigma_0 \left(W_0 \boldsymbol{y} + \boldsymbol{b}_0 \right) + \boldsymbol{b}_1 \right) + \boldsymbol{b}_2 \right) \dots + \boldsymbol{b}_n \right)$$

where the input is repeatedly multiplied by a weight matrix W_i and passed through a (nonlinear) activation function σ, i.e. $\boldsymbol{\theta}$ contains $W_{0,\dots,n}$, $\boldsymbol{b}_{0,\dots,n}$, where $W_i \in \mathbb{R}^{s_i \times s_{i-1}}$ are real weight matrices, $\boldsymbol{b}_i \in \mathbb{R}^{s_i}$ are bias vectors, and σ_i are activation functions (e.g. sigmoid or tanh), applied elementwise. If all σ_i are linear, then the resulting mappings are linear. The shape of each weight matrix W_i determines the "width" of the subsequent layer.

We can then try to find a parametrization $\boldsymbol{\theta}$ that minimizes the triplet loss function 2, e.g. by stochastic gradient descent.

[JHF: Include results]

Challenges:

- Practically, optimizing $\boldsymbol{\theta}$ may be difficult when the number of layers n is large, due to "vanishing gradients" and other problems induced by correlations across layers.
- A MLP has an exponential number of symmetrical optima due to parameter unidentifiability. This is not a problem if we want to just find one good setting of the parameters, but may be important to keep in mind if we are building a

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 this projection function is invertible, then this now lets us infer the latent coordinates given observations.

Probabilistic PCA (PPCA)

We can reformulate PCA in probabilistic terms by constructing a latent variable model, where the prior over hidden coordinates is Gaussian, and the high-dimensional observations $\{\boldsymbol{y}_t\}$ are linear projections of hidden coordinates $\{\boldsymbol{x}_t\}$ corrupted by isotropic Gaussian noise (i.e. $\boldsymbol{y}_t \sim M\boldsymbol{x}_t + \mathcal{N}(\mathbf{0}, \mathbf{I}\sigma^2)$). PCA can then be interpreted as the maximum likelihood estimate of this linear projection function M .[7] [8]

Gaussian Process Latent Variable Model (GP-LVM)

Nonlinear generalization of PPCA, where the latent-to-observed projection functions are modeled as draws from a GP. Reduces to PPCA when the GP kernel is linear.

DYNAMICS MODELS

Markov State Models (MSMs)

We assume the data sequence $\{\boldsymbol{y}_t\}$ is generated by an ergodic, reversible Markov jump process over discrete clusters.

Projected and Hidden Markov Models (PMMs and HMMs)

In an HMM, we assume there is an ergodic, reversible Markov jump process $\{\boldsymbol{x}_t\}$ over a discrete hidden state space $\boldsymbol{x}_t \in \mathbb{Z}^M$, representable by an $M \times M$ transition matrix. The observed sequence $\{\boldsymbol{y}_t\}$ over a discrete observable space $\boldsymbol{y}_t \in \mathbb{Z}^N$ is given by a discrete conditional distribution, $p(\boldsymbol{y}_t|\boldsymbol{x}_t)$, representable by an $M \times N$ observation matrix.

In the case of PMMs, we assume there is an ergodic, reversible Markov process $\{x_t\}$ over a continuous hidden state space $\boldsymbol{x}_t \in \Omega$. The observed sequence $\{y_t\}$ are projections of $\boldsymbol{x}_t$.

PMMs can be approximated by HMMs, and there are efficient algorithms for learning the parameters of HMMs. Additionally, any observable that can be estimated from an MSM can be estimated from a PMM / HMM.

[JHF: Reference: http://publications.mi.fu-berlin.de/1320/]

[7] Or, more precisely, of its inverse M^T.

[8] Note that this, like PCA, ignores the time-ordering of the observations.

Reduced-Rank Hidden Markov Models (RR-HMMs)

If we assume that the hidden transition matrix is rank-deficient (i.e. it has rank $m \ll M$), then we can model processes involving a very large number of hidden states M while only having to estimate $m \times m$ parameters.

Linear dynamical systems (LDS)

The time propagator is a linear function of the current state.

Gaussian Process Dynamical Model

Generalization of linear dynamical systems, where the propagator is modeled by a GP.