Interface for Variational Package

July 13, 2015

Here, we briefly sketch the interface for all functions to appear in the variational package. The package consists of three main modules: A library of basis sets, estimators for the correlation matrices, and a solver for the resulting generalized eigenvalue problem.

1. The basis sets library contains functions to evaluate specific classes of basis functions. Examples for these classes are Gaussian basis functions, Fourier waves defined on angles, or the MSM-eigenvector based functions and their products. The general interface is given by the function SomeBasisSet below.

2. The estimator module contains a function that generates the correlation-matrices $\mathbf{C}^{\tau}$, $\mathbf{C}^{0}$ from the basis function trajectories generated in the first step.

3. The solver module contains a function the solve the generalized eigenvalue problem for the correlation matrices generated before. We will just use the function eig_corr implemented in pyemma.util.linalg.

Below we describe the interfaces for these three modules.

```
def SomeBasisSet(list_of_trajectories, prefix, parameters
    ):
    """
    Parameters
    ----------
    list_of_trajectories: list
        List of .npy-files. Each file contains a feature
            trajectory, represented as an np-array of
            shape (T,N), where T is the number of time-
            steps in this trajectory and N is the number
            of features (distances, angles,...) on which
            the basis set is defined.
    prefix: string
        Common prefix for all files to be produced (see
            Output).
    parameters:
        Additional parameters needed for this basis set.

    Returns
    -------
        Returns a list of lists of filenames where the
            evaluations of all requested basis functions
            can be found. The files will be called "
            prefix_trajnum_fctnum.npy", where trajnum is
            the trajectory number and ftcnum is the number
             of the basis function. Each sublist contains
            the files for one trajectory.
    '''
```

```
def Estimator(list_of_trajectories, list_of_taus):
    """
    Parameters:
    ------------

    list_of_trajectories: list
        List of list of .npy-files, organized the same
            way as the output of a basis set function.
            Each sublist contains the files for all basis
            functions for one specific trajectory.
    list_of_taus: ndarray (ntau,)
        The lag-times for which the correlation matrices
            will be computed.

    Returns
    --------

    list of correlation matrices.
    """



def eig_corr(C0, Ct, epsilon=1e-6):
    """      Solve the generalized eigenvalues problem
        with correlation matrices C0 and Ct
    Parameters
    -----------

    C0 : ndarray (n,n)
        time-instantaneous correlation matrix. Must be
            symmetric positive definite
    Ct : ndarray (n,n)
        time-lagged correlation matrix. Must be symmetric
    epsilon : float
        eigenvalue norm cutoff. Eigenvalues of C0 with
            norms <= epsilon will be cut off. The
            remaining number of Eigenvalues define the
            size of          the output.
    Returns
    --------

    l : ndarray (m)
        The first m generalized eigenvalues, sorted by
            descending norm
    R : ndarray (n,m)
        The first m generalized eigenvectors, as a column
             matrix.
    """
```