Optimizations python

July 2019

1 Introduction

1.1 Matrix operations

1.1.1 Power Iteration

The ACDC algorithm requires the computation of the largest eigenvalue and its associated eigenvector of some matrix A. Usually, computing the eigensystem is done by eigenvalue decomposition, which includes solving the characteristic polynomial and provides every eigenvalue and eigenvector of the matrix. This allows the largest eigenvalue and its eigenvector to be selected.

In stead of computing the complete eigenvalue decomposition, it can be faster (for large matrices) to perform the power iteration algorithm to approximate only the eigenvector corresponding to the largest eigenvalue. Starting from a random vector v_0, repeat until convergence:

$$v_{i+1} = \frac{Av_k}{||Av_k||} \tag{1}$$

If A has a unique largest eigenvalue and the starting vector has a non-zero component in the direction of the largest eigenvector, then a subsequence v_i converges to the eigenvector associated with the largest eigenvalue.

1.1.2 Rayleigh Quotient

For a given real symmetric matrix A and non-zero vector v, the Rayleigh quotient is defined as

$$RQ(A, v) = \frac{v^T Av}{v^T v} \tag{2}$$

The Rayleigh quotient reaches it maximum value $RQ(A, v) = \lambda_{max}$, the largest eigenvalue of A, when v is the corresponding eigenvector.

1.1.3 Einstein Summation Convention

The Einstein summation convention is a notational convention that implies the summation over indexed terms in a formula. Not only does this achieve notational brevity, but it can be used to combine multiple steps of computation into one.

Since the introduction of the function einsum in the Python package numpy up untill at least numpy version 1.14.0, einsum provides some advantages in memory accessing that allow some computations, like computing the outer product, to be faster than using numpy's built in function. In the joint diagonalization computations, the outer product of two vectors is computed using einsum.

1.2 Cross-correlation

The matrix of cross-correlations of the measured (whitened) signal $X(t)$ at time-lag τ is defined as

$$R(\tau) = \mathbb{E}[X(t)X(t-\tau)] = \int_{-\infty}^{\infty} X(t)\overline{X}(t-\tau)\,\mathrm{d}t \tag{3}$$

1.2.1 Data Whitening

Whitening of data X is performed by finding the transformation matrix U such that $X_{white} = UX$ results in X_{white} having unit diagonal co-variance. A commonly used choice for U is the eigen-system of the co-variance matrix associated with X. The eigenvectors in U rotate X along an axis that ensures the unit co-variance. Alternatively, by directly calculating the singular value decomposition, a factorization of X, we obtain

$$U\Sigma V = X \tag{4}$$

The columns of U and V are the left-singular values and right singular vectors of X respectively. The left singular vectors of X are a set of orthonormal eigenvectors of XX^T, and the right singular vectors are a set of orthonormal eigenvectors of X^TX. Since U and V are both orthonormal, the whitened matrix can be obtained by

$$UV = X_{white} \tag{5}$$

1.2.2 Fourier Transform Convolution

The cross-correlation over all lags (from -nsamples to +nsamples) can be computed using the Fourier transform convolution. For two signals x_1 and x_2 the cross-correlation is defined as

$$xcorr(\tau) = \sum_{t=0}^{|x1|-1} x_1(t)x_2^*(t-\tau+|x_1|-1) \tag{6}$$

where $x_2(t) = 0$ when t is outside the range of x_2. This is equal to computing the convolution of the two signals as $(x_1 * x_2)(\tau - |x_1| + 1)$.

Fourier transform convolve is generally very efficient for computing cross-correlations. However, when the signal is very long and only a few lags are needed, a lot of unnecessary computations are done.

1.2.3 Calculating Cross-correlation Directly

The matrix of cross-correlations $R(\tau)$ of the whitened signal $X(t)$ can be computed directly for each τ. The signal is first shifted by the current τ, and then the expected value of the matrix product is computed. Instead of computing the last two steps separately, these steps are combined into one computation using Einstein notation. When only a few lags are needed, this proves to be the faster method for computing cross-correlation.