

1 1/24/16

1.1 Permutation testing

In order to test how the algorithm returns the permutation results, I set up sample matrices using `create_toy.m` and recorded the distance between the result returned by the algorithm (`eig_perm`) and that of the actual input data (`ss`).

Listing 1: Compare Permutations

```
rois = 33;
noise_mag = 0;
trace_type = 'sines';
shift = 10;
y = zeros(1,100);
for i = 1:length(y)
    [Z, ss] = create_toy(trace_type, 'rois', rois, ...
        'noisemag', noise_mag, ...
        'shift', shift);
    [eig_phases, eig_perm, slm, evals] = cyclic_analysis(Z);
    y(i) = cyclic_distance(ss, eig_perm);
end
plot(y)
```

The function `create_toy` produces a set of `rois` identical sine waves randomly spread over a `shift` time step period (i.e., each copy of the trace is shifted along the x axis so that all copies are within `shift` time steps of each other). The variable `ss` shows the ordering of the traces produced. Using the setting above, for example, a data set like that in on the left of Figure 1 is produced. The figure on the right shows that the algorithm returns the identical permutation each time. The distance is measured using `cyclic_distance(V1,V2)` which simply counts the how many steps V2 is from V1 (let V1 be (1:5), then [2,1,3,4,5] is a distance of 1 from V1 and [2,3,4,5,1] is a distance of 5 from V1.) This doesn't really count "cyclic distance" but works for this application since we would like to see if we are picking up the correct starting point and cycling in the right direction. Increasing `shift` to 18 (the period for the sine wave is 20) starts to mess up the permutation since the algorithm can't determine the proper starting point.

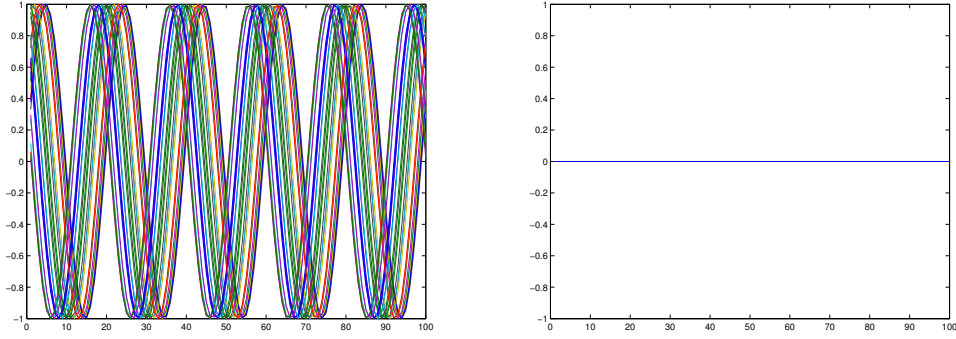


Figure 1: Left: Sample data created from sine waves shifted randomly at most 10 time steps apart (no added noise). Right: Comparison of the permutation returned by the algorithm with that of the actual data set for 100 separately generated data sets.

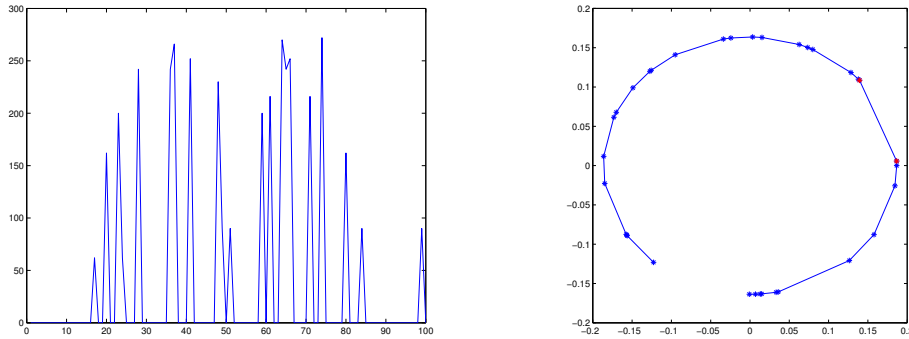


Figure 2: Left: Algorithm performance when `shift=18`. Right: Plot of phases for a permutation which the algorithm had shifted by 11 places (distance = 242). The red asterisks highlight the 11th and 12th points in the cycle.

The algorithm determines the starting point by finding the largest gap between points in the phase plot (greatest difference in angle). Once the phases are too spread out, the largest gap can easily be between a pair of middle points. Note, however, that still the algorithm cycles in the correct direction in each case.

2 Listings

Listing 2: `cyclic_distance`

```
function [dist] = cyclic_distance(V1, V2)
[~, ss] = sort([V1(:), V2(:)]);
dist = sum(abs(diff(ss,1,2)))/2;
```