Algorithms for Spectral vs Correlational Discriminability Approach

ericwb95

February 2017

Algorithm 1: Amplitude Spectrum

Input:
$signal \in \mathbb{R}^{t,r}$ the timeseries.
$normalize$ a bool indicating whether you want each spectrum to sum to one per roi.

Result:
$amp \in \mathbb{R}^{t/2,r}$ the amplitude spectrum.

for $r \in 1 : dim(signal)[2]$ **do**
  $ampsig = fft(signal[,r])$
  $ampsig = 2 * abs(ampsig[1 : t/2,])$
  if $normalize$ **then**
    $ampsig = ampsig/sum(ampsig)$
  $amp[,r] = ampsig$
end
return amp

Algorithm 2: Power Spectrum

Input:
$signal \in \mathbb{R}^{t,r}$ the timeseries.
$normalize$ a bool indicating whether you want each spectrum to sum to one per roi.

Result:
$pow \in \mathbb{R}^{t/2,r}$ the power spectrum.

for $r \in 1 : dim(signal)[2]$ **do**
  $powsig = fft(signal[,r])$
  $powsig = abs(powsig[1 : t/2,])^2$
  if $normalize$ **then**
    $powsig = powsig/sum(powsig)$
  $pow[,r] = powsig$
end
return pow

Algorithm 3: Divergence

Input:
$P \in \mathbb{R}^t$ the timeseries for a single ROI.
$Q \in \mathbb{R}^t$ the timeseries for a second ROI.

Result:
$div \in \mathbb{R}$ KL-divergence.

$div = \sum_{i=1}^{t} P(i) \log \left(\frac{P(i)}{Q(i)} \right)$
return div

Algorithm 4: Divergence

Input:

$sig \in \mathbb{R}^{t,r}$ an ROI timeseries.

Result:

$divmx \in \mathbb{R}^{r,r}$ matrix representing edge-wise divergences between ROI timeseries.

for $r1 \in 1 : dim(sig)[2]$ **do**
  for $r2 \in 1 : dim(sig)[2]$ **do**
    $divmx[roi1, roi2] = divergence(sig[, r1], sig[, r2])$
  end
end
return $divmx$

Algorithm 5: Hellinger Distance

Input:

$P \in \mathbb{R}^{r,r}$ the similarity matrix for a single subject.

$Q \in \mathbb{R}^{r,r}$ the similarity matrix for a second subject.

Result:

$hdist \in \mathbb{R}$ The hellinger distance between two matrices.

$hdist = \frac{1}{\sqrt{2}} \left|\left|\sqrt{P} - \sqrt{Q}\right|\right|_F$
return $hdist$

Algorithm 6: Dataset Edge-Wise Distance

Input:

$g \in \mathbb{R}^{s,r,r}$ a collection of graphs for the subjects in our dataset.

Result:

$dist \in \mathbb{R}^{s,s}$ matrix representing edge-wise distances between pairs of graphs.

for $s1 \in 1 : dim(g)[1]$ **do**
  for $s2 \in 1 : dim(g)[1]$ **do**
    $dist[s1, s2] = hdist(g[s1, ,], g[s2, ,])$
  end
end
return $divmx$