Cointegration Kernel with Tom Li and Paul Lintilhac

In this project, we define a "Cointegration Kernel" that can be used to analyze time-series that are "causally related". The overall motivation for this research is to give an alternative approach for analyzing large numbers of time series. The basic setting of such a problem might be like this: you are working at a company with a large amount of private data in the form of time-series, and you want to know if any of your time series might be predictive of important external economic indicators, financial indexes, or any other potentially valuable signals. The classic approach to this would be to look for high leading and lagging correlations between any of your internal time series and any external timeseries.

While this problem in general suffers from extremely high computational complexity when there are a large number of time-series involved, it can become more tractable and flexible when viewing the correlation as a kernel.

Example 1: correlation kernel

Given a set of non-constant time series of length N, $X \in \mathbb{R}^N \backslash \{1, \ldots ,1\}\alpha \colon \alpha \in \mathbb{R}$

Let us define the feature map

$$\Phi \colon X \rightarrow X$$

$$\Phi(x) = \frac{x - \mu_x}{\sigma_x}$$

i.e., Φ de-means and normalizes the time series in X. Then we can write the correlation between $x, y \in X$ as the inner product

$$C(x, y) =< \Phi(x), \Phi(y) >$$

Note that in addition, because the timeseries are normalized and de-meaned by Φ, we can express this correlation in terms of a distance:

$$\left(\Phi(x) - \Phi(y)\right)^2 =< \Phi(x), \Phi(x) > + < \Phi(y), \Phi(y) > -2 < \Phi(x), \Phi(y) >$$

$$= 1 + 1 - 2 < \Phi(x), \Phi(y) >$$

$$= 2 - 2C(x, y)$$

Now, since the LHS is a known PDS kernel, the RHS is as well. Thus we can construct a PDS kernel using a simple function of the correlation, allowing us to do useful things with this kernel, such as SVM or hierarchical clustering.

Example 2:

If we wanted to extend this kernel function to cross correlations, we might ask not for a simple correlation between two time series, but the maximum cross correlation over all possible lags, γ, i.e.

$$CC(x, y) = \max_{\gamma} < \Phi(x)_{\gamma}, \Phi(y) >,$$

Or equivalently,

$$CC(x,y) = \min_{\gamma}\{2 - 2 < \Phi(x)_{\gamma}, \Phi(y) >\}$$

Where $x_{\gamma} = x_{t-\gamma}$ is the time series x shifted back by γ.

Now, note that since $2 - 2 < \Phi(x)_{\gamma}, \Phi(y) >$ is PDS, $2 < \Phi(x)_{\gamma}, \Phi(y) > \; -2$ is PDS. Then let us replace the hard minimum shown above with a soft minimum, defined as

$$CCS(x,y) = \sum_{\gamma=0}^{\gamma_{max}} e^{-2<\Phi(x)_{\gamma},\Phi(y)> +2}$$

Since -CC(x,y) is NDS, each term in the sum is a Gaussian kernel and is therefore PDS, which means that the entire sum is PDS.

Comments of efficiency:

Note that we can write the cross correlation for all γ as the discrete convolution

$$CC(x,y,\gamma) = \sum_{i=1}^{N} x_i y_{i-\gamma}$$

Interestingly, it can be shown that this convolution can be done in terms of an inner product in Fourier space with one of the timeseries reversed, which allows us to complete to computation in N logN time, rather that N^2 time using FFT. Therefore we have

$$CC(x,y,\gamma) = \mathscr{F}^{-1}\left(\mathscr{F}(\Phi(x)) \cdot \mathscr{F}(\Phi(-y))\right)$$

In recently released white-papers, Google has demonstrated the use of multi-level clustering techniques using this correlation kernel in their Google Correlate product in order to pre-process the time series so that querying for nearest neighbors is much more efficient. Perhaps this same technique could be combined with the efficient cross correlation algorithm above to make this clustering feasible even with cross-correlations.

Project: Cointegration Kernel

However, even with all of these efficiency gains, we note that if we are under the assumption of a large number of time-series, then no matter how efficient our algorithms are, there is a very significant chance that any highly correlated time-series could be spurious correlations. Therefore we may want to impose an additional constraint that the time-series be co-integrated. Often time series that are not only correlated but also cointegrated are considered to have a stronger causal relationship than two time-series with the same correlation but no co-integration. This is because their relationship appears to have some “memory”, in that a certain linear combination of the time-series is error-correcting.

The classical test to see whether two time-series normalized and de-meaned time-series x and y are cointegrated is the Engle-Granger, which consists of two parts:

1) Regress x on y using least-squares, and then take the residual. We can define this residual explicitly using the normal equations:

$$\varepsilon_{XY,t} = y_t - x_t \boldsymbol{x^T y}$$

2) Next, we run a unit-root test on ε_{XY} in order to determine if it is stationary. If it is, then we say the two series X and Y are cointegrated. This usually takes the form of an Augmented Dickey-Fuller Test. In our case, we opt for a straightforward calculation of the autocorrelation with lag γ:

$$K_{XY}(\gamma) = < \varepsilon_{XY,t} \cdot \varepsilon_{XY,t-\gamma} >$$

$$= < y_t - x_t \boldsymbol{x^T y}\, , y_{t-1} - x_{t-1} \boldsymbol{x^T y} >$$

Now the issue with this is that it is not symmetric. But we can symmetrize the kernel as follows:

$$K(x, y, \gamma) = \frac{1}{2}\left(K_{XY}(\gamma) + K_{YX}(\gamma)\right)$$

Now C is a kernel because it is symmetric. The interpretation is that we are now looking at serial correlations of the y-residuals of Y regressed on X as well as serial correlations in the x-residuals for X regressed on Y. It makes intuitive sense that if the series are really reverting back to some trend line, that both the x and y residuals should be O(1), i.e stationary.

Using the same general argument as in the examples above, we can create an NDS kernel using this kernel by observing the fact that it is just a correlation, i.e. $-1 \leq C \leq 1$. Therefore we can define a notion of distance again by taking $D(x, y, \gamma) = 2 - 2K(x, y, \gamma)$ and observe that this kernel is NDS. Therefore we can again define a PDS kernel as:

$$KS(x, y) = \sum_{\gamma=0}^{\gamma_{max}} e^{2-2K(X,Y,\gamma)}$$

This is a PSDS kernel representing a soft maximum of the level of co-integration between the two time-series.

Questions:

- Can we express process as an inner product of certain features of X and Y which are symmetric, i.e. such that our kernel

$$C = \Phi(\boldsymbol{X}) \cdot \Phi(\boldsymbol{Y})?$$

$$(answer\ so\ far\ is\ no, but\ its\ certainly\ possible)$$

- Is this kernel PDS? I believe the above argument shows it is.