AR model using Yule-Walker method

Matti Pastell

14.5.2013

```
from scipy import signal, linalg
import numpy as np
import matplotlib.pyplot as plt

class YW(object):
    """A class to fit AR model using Yule-Walker method"""

    def __init__(self, X):
        self.X = X - np.mean(X)
```

1 Calculate autocorrelation

YW method requires that we compute the sample autocorrelation function:

$$r_k = \frac{1}{(n-k)\sigma^2} \sum_{t=1}^{n-k} (X_t - \mu)(X_{t+k} - \mu)$$

```
    def autocorr(self, lag=10):
        c = np.correlate(self.X, self.X, 'full')
        mid = np.floor(len(c)/2)
        acov = c[mid:mid+lag]
        acor = acov/acov[0]
        return(acor)
```

2 Fit

Form the Yule-Walker equations $r = R\Phi$ based on sample autocorrelation r_k. Notice that the matrix R is a Toeplizt matrix and it is thus easy to form using `toeplitz` function from `scipy.linalg`.

$$\begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_p \end{pmatrix} = \begin{pmatrix} r_0 & r_1 & \dots & r_{p-1} \\ r_1 & r_0 & \dots & r_{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ r_{p-1} & r_{p-2} & \dots & r_0 \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_p \end{pmatrix}$$

And solve simply using:

$$\Phi = R^{-1} r$$

```
    def fit(self, p=5):
        ac = self.autocorr(p+1)
        R = linalg.toeplitz(ac[:p])
        r = ac[1:p+1]
        self.phi = linalg.inv(R).dot(r)
```

3 Calculate and plot the spectrum

The spectrum of an AR process is given by:

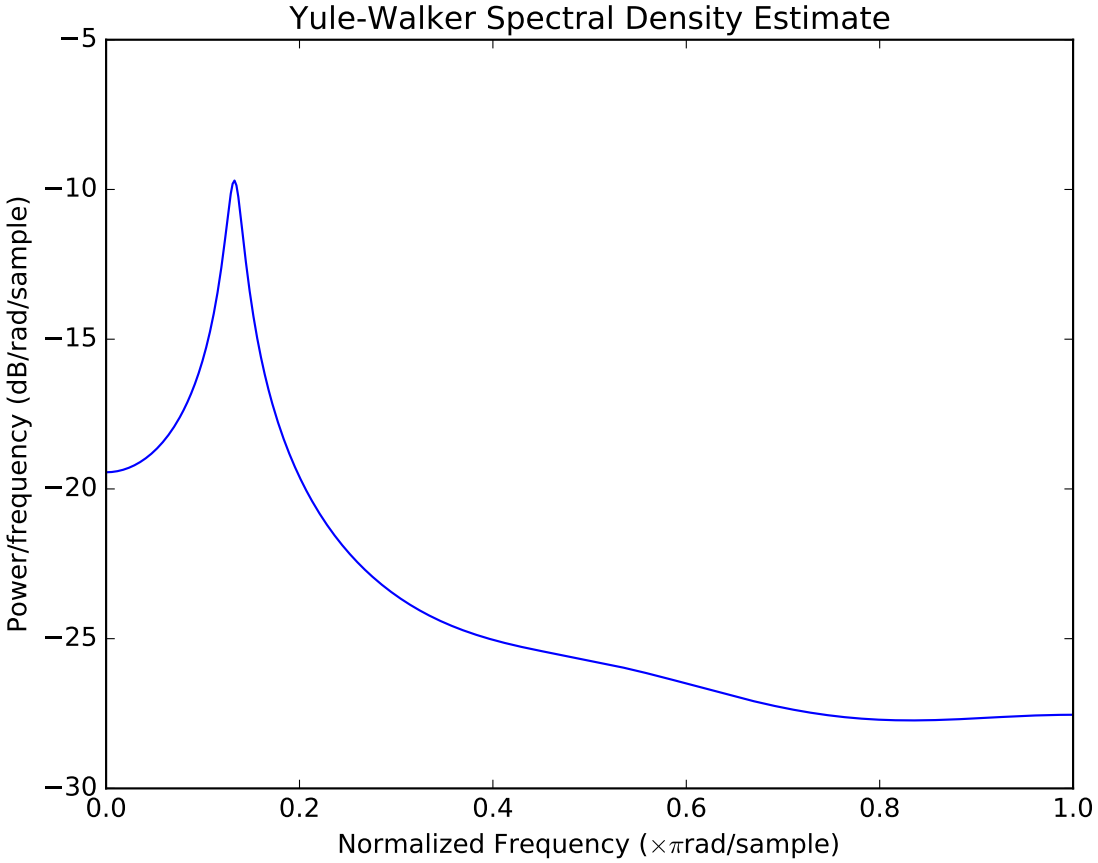
$$S(f) = \frac{\sigma^2}{|1 - \sum_{k=1}^{p} \phi_k e^{-2\pi i k f}|^2}$$

It can be calcuted easily using `scipy.signal.freqz`.

```
    def spectrum(self):
        a = np.concatenate([np.ones(1), -self.phi])
        w, h = signal.freqz(1, a)
        h_db = 10*np.log10(2*(np.abs(h)/len(h)))
        plt.plot(w/np.pi, h_db)
        plt.xlabel(r'Normalized Frequency ($\times \pi$rad/sample)')
        plt.ylabel(r'Power/frequency (dB/rad/sample)')
        plt.title(r'Yule-Walker Spectral Density Estimate')
```

4 Try it out:

```
>>> x = np.sin(np.linspace(0, 20))
>>> ar1 = YW(x)
>>> ar1.fit()
<string>:14: VisibleDeprecationWarning: using a non-integer number
instead of an integer will result in an error in the future
>>> ar1.phi
array([ 1.19379795, -0.21810471, -0.12747881, -0.06257484,
-0.12929761])
>>> ar1.spectrum()
```


Yule-Walker Spectral Density Estimate
Power/frequency (dB/rad/sample)
−5
−10
−15
−20
−25
−30
0.0
0.2
0.4
0.6
0.8
1.0
Normalized Frequency (×πrad/sample)