Time-series and scaling analysis with R

Steve UHLIG

Generate heavy-tailed distribution

- library(VGAM)
- Generating Pareto distribution:
  - vector.pareto1 <- rpareto(n=100000,location=10,shape=1)
  - vector.pareto1.5 <- rpareto(n=100000,location=10,shape=1.5)
  - vector.pareto2 <- rpareto(n=100000,location=10,shape=2)
- Plot CCDF on a log-log scale:
  - ```
    ccdf <- function(myList,density=FALSE)
    {
        # generates the CCDF of a list or vector
        freqs = table(myList)
        X = rev(as.numeric(names(freqs)))
        Y =cumsum(rev(as.list(freqs)));
        data.frame(x=X,count=Y)
    }
    ```
  - plot(ccdf(vector.paretox),log=“xy”)
    ```

Time-series aggregation

- library(tseries)
- Computing m-aggregated time series:
  - vector.ts <- ts(vector,frequency=m)
  - vector.m-agg <- aggregate(vector.ts,FUN=mean)
- Plot aggregated time series:
  - plot(vector.m-agg)
- Compare variances of aggregated time-series:
  - poisson decreases proportionately with m
  - heavy-tailed very slowly

Auto-correlation

- library(stats)
- Auto-correlation function: acf()
  - Poisson process is uncorrelated: 1 at lag 0, 0 otherwise
  - Pareto time-series is also uncorrelated: 1 at lag 0, 0 otherwise
  - Real traffic is correlated: acf(vector,log=“y”) shows slow decay

Wavelets

- library(waveslim)
- Performing wavelet analysis:
  - vector.res <- dwt(vector [1:2**16],wf="d16",n.levels=10,boundary="periodic")
  - Length of analyzed time-series (L) must be a power of 2
  - Several wavelet functions available (wf)
  - Number of levels to be analyzed: 2^n <= L

Wavelets

- Selecting wavelet coefficients at scale i
  - vector.res$di
- Computing log2 of their variance:
  - log2(var((res.vector$di)))
- Keeping energy of coefficients at each scale:
  - coefvar <- mat.or.vec(9,1)
  - coefvar[i] <- log2(var((res.vector$di)))
- Plot the log of the energy of the wavelet coefficients
  - Poisson process shows no scaling
  - Real traffic does