

Documentation of MATLAB functions, scripts and data sets

Florian Nisbach

26th September 2006

Contents

1	Changed Functions	2
1.1	cumprobdist_ng.m	2
1.2	degree_stw.m	2
1.3	simulateTimeWindows.m	3
1.4	timewin3d.m	3
2	Created Functions	3
2.1	connectionHyp.m	3
2.2	corrtest.m	4
2.3	degree_stat.m	4
2.4	degrees.m	5
2.5	distri.m	5
2.6	fitplot.m	5
2.7	plot_chresults.m	6
2.8	plot_results.m	6
2.9	prepareParmVector.m	6
2.10	ptimepoly.m	6
2.11	randomgraph_symm.m	7
2.12	rgcalc.m	7
2.13	sigma.m	7
2.14	sigma.m	7
3	Scripts	8
3.1	calcswfactors.m	8
3.2	degree_script.m	8
3.3	degree_script.m	8

3.4	script_connectionHyp.m	8
3.5	script_connectionHyp.m	9
4	Generated Datasets	9
4.1	data_connectionHyp2006_08_26.mat	9
4.2	quantiles_data_2006_09_04.mat	9
4.3	quantiles_data_2006_09_18.mat	9
4.4	SW_08-15-dataset.mat	9
4.5	SW_08-18-dataset.mat	9
4.6	timewin_data_08_15.mat	9
4.7	timewin_data_08_18.mat	9

1 Changed Functions

1.1 cumprobdist_ng.m

```

% [r,lambda] = cumprobdist_ng(dl)
% same as cumprobdist, only without graphical output
% given the degree list, calculates the
% cumulative distribution
% and yields linear fit correlation
% coefficient r and (negative) steepness
% lambda of linear fit.
% Reference: Yook2002 for cumulative distribution
%           Newman2001 for uniform width binning
% Author: Marcus Kaiser   Date: 8.12.2002
% changed by:      Florian Nisbach
% Last change:    2006-08-15

```

The graphical output was removed from the original cumprobdist.m to include it into batch calculations.

1.2 degree_stw.m

```

% function [quantiles,dd_r,dd_lambda,runtime] =
% degree_stw(nodes, windows, ParmVector);
% yet another version of simulateTimeWindows,
% which stores all percentiles
% of the degree distribution, as well as fit
% data for testing for power law
% distribution.

```

```
% based on timewin3d by Marcus Kaiser
% Author:          Florian Nisbach
% Date of creation: 2006-08
% Last change:     2006-09-14
```

Used by degree_stat.m.

1.3 simulateTimeWindows.m

```
% [dd_r,dd_lambda,wiring_r,wiring_lambda,wiring_total,CC,ASP,Edges] =
% simulateTimeWindows(nodes, windows, ParmVector)
% Part of a modularized version of timewin3d, to do statistics...
% Original Author:  Marcus Kaiser
% Changed by:       Florian Nisbach
% Last change:     2006-08-18
```

Adapted version of timewin-3d, with graphical output removed and a few bugfixes

1.4 timewin3d.m

```
% function [matrix,position,matrixc,d,cl,shp] =
% timewin3d(nodes,windows);
% Program:      timewinspatialgrowth(param) - spatial network growth
%               using time windows and different node classes
%               (graph coloring)
% Author:       Marcus Kaiser
%               International University Bremen, Germany
% Changed by:   Florian Nisbach
% Date:         31 Jan 2006 (14 Jun 2003)
% Last change:  2006-08-17
```

The time window functions were changed to the polynomial based ones. Some errors were removed.

2 Created Functions

2.1 connectionHyp.m

```
% function [EdgeDist, InterEdgeDist] =
% connectionHyp(nodestep,numsteps, windows, ParmVector);
```

```

% Stores the timed distribution of total edges
% and inter-class edges.
% nodestep is the stepwidth, numsteps is the number of steps
% (so that NODES = nodestep*numsteps)
% windows is the number of time windows
% ParmVector is a suitable parameter vector for the time
% window functions
% Author:          Florian Nisbach
% Date of creation: 2006-08
% Last change:     2006-08-25

```

This function, based on `timewin-3d.m`, saves the development of edges over the runtime.

2.2 `corrtest.m`

```

function corrtest(A_Data, axis, tw_type)
% function corrtest(A_Data, axis, tw_type)
% tests the correlation of A_Data (A_ASP and so on, as given
% from stat_script)
% axis is one of 'nodes', 'windows', 'integral'
% tw_type is one of 'all', 'theonly', 'nothree'
% Author:          Florian Nisbach
% Date of creation: 2006-09-13
% Last change:     2006-09-13

```

The input is a 3-dimensional data array. One axis is selected, and the mean over the other two is calculated, and it can be specified if the three-time-window case shall be included, omitted or dealt with exclusively. The correlation coefficient of the values over the axis is calculated, as well as the r^2 values of the significance for the following function fits: Polynomials of degrees 1 to 4; exponential; Power law; Power law plus a constant. The data points are plotted.

2.3 `degree_stat.m`

```

% function out = degree_stat(nodes, windows, Integral, numruns)
% runs degree_stw several times and to calculate mean
% and estimate for standard deviation.
% Author:          Florian Nisbach
% Date of creation: 2006-08
% Last change:     2006-09-14

```

The output is a record, used by `degree_script.m`.

2.4 `degrees.m`

```
% dl = degrees(matrix)
% list of degree for each node
% (degree distribution)
% Original version: Marcus Kaiser
% Author (rewrite): Florian Nisbach
% Date of creation: 2006-08
% Last change:      2006-09-14
```

Original version produced errors, as the adjacency matrices generated by `timewin_3d.m` use the values of the non-zero elements to store node classes. Complete rewrite, also with counting the diagonal elements twice error fixed.

2.5 `distri.m`

```
% function dist=distri(windows,nodes,alpha,A)
% pics out a distribution corresponding to the
% given parameter vector from A, which should be
% the 4D percentiles array generated by
% degree_script.m
% Author:          Florian Nisbach
% Date of creation: 2006-09
% Last change:     2006-09-11
```

2.6 `fitplot.m`

```
% function fitplot(A_Data, axis, tw_type, model)
% plots a fitted function through collapsed A_Data (see corrtest)
% axis is one of 'nodes', 'windows', 'integral'
% tw_type is one of 'all', 'theonly', 'nothree'
% model is fit model, see cflibhelp for help
% Author:          Florian Nisbach
% Date of creation: 2006-09-13
% Last change:     2006-09-13
```

Plots the actual fitted function through the data points. Second half of the toolkit provided together with `corrtest.m`.

2.7 plot_chresults.m

```
% function plot_chresults(A,tit)
% plot results for the connection hypothesis script
% Author:          Florian Nisbach
% Date of creation: 2006-09
% Last change:     2006-09-14
```

Plots the results of the timed edges growth array as a nice colorful slice plot.
See plot_results.m.

2.8 plot_results.m

```
% function plot_results(WINDOWLIST,NODESLIST,INTLIST,A,tit)
% plot results for stat_script and degree_script
% Author:          Florian Nisbach
% Date of creation: 2006-08
% Last change:     2006-09-14
```

Plots the results as a nice colorful slice plot.

2.9 prepareParmVector.m

```
% function musigma = prepareParmVector(Windows, Integral);
% calculates the mu and sigma vector for the time window functions
% (TWF), taking the number of windows and the desired value of the
% integral over the TWFs over the interval [0,1].
% Author:          Florian Nisbach
% Date of creation: 2006-08
% Last change:     2006-08-14
```

Used in TWstat.m and others. This was put in an extra function to save calculation time, because it used to be slow.

2.10 ptimepoly.m

```
% function p = ptimepoly(x,mu,sigma)
% This function calculates the polynomial based time
% window functions, depending on mu and sigma
% Author:          Florian Nisbach
% Date of creation: 2006-08
% Last change:     2006-08-15
```

2.11 randomgraph_symm.m

```
% function G = randomgraph_symm(n,e)
% generates the adjacency matrix of a symmetric random
% graph with n nodes and e edges
% Author:          Florian Nisbach
% Date of creation: 2006-08
% Last change:     2006-08-17
```

2.12 rgcalc.m

```
% function [ASP, CC] = rgcalc(nodes, edges, runs)
% calculates the average ASP and the CC of a random graph
% generated with randomgraph_symm
% Author:          Florian Nisbach
% Date of creation: 2006-08
% Last change:     2006-08-17
```

2.13 sigma.m

```
% function s = sigma(mu, I)
% compute the exponent sigma for the time window function ptimepoly
% NOTE: mu (x-value of the peak) and I (desired integral) must be
% between 0 and 1!
% Author:          Florian Nisbach
% Date of creation: 2006-08
% Last change:     2006-08-14
```

The computations are done numerically. For integration, the built-in function `quadv` is used. For finding the root, an interval subdivision algorithm is used, for the sake of numerical stability.

2.14 sigma.m

```
% function out = TWstat(nodes, windows, Integral, numruns)
% runs simulateTimeWindows several times, calculating the mean and
% estimated standard deviations for the measured values. The parameter
% vector for the time window functions used is pre-calculated.
% The output is stored in an array.
% Author:          Florian Nisbach
% Date of creation: 2006-08
% Last change:     2006-08-18
```

Used by stat_script.m.

3 Scripts

3.1 calcswfactors.m

```
% Calculate SW-Factors of the dataset
% Author:          Florian Nisbach
% Date of creation: 2006-08-23
% Last change:     2006-08-23
```

This script was used to calculate ASP_{gen}/ASP_{rand} and CC_{gen}/CC_{rand} with a pre-calculated dataset generated by stat_script.m

3.2 degree_script.m

```
% script to run statistics using degree_stat
% the percentile data is stored in the 4D array A_quantiles
% r and lambda of degree distribution is stored,
% as well as runtime
% Author:          Florian Nisbach
% Date of creation: 2006-08
% Last change:     2006-09-14
```

3.3 degree_script.m

```
% Plots the polynomial based time window functions
% used in all simulations.
% Author:          Florian Nisbach
% Date of creation: 2006-08
% Last change:     2006-09-15
```

3.4 script_connectionHyp.m

```
% this script was used to calculate the edges development over
% the time of network generation.
% Author:          Florian Nisbach
% Date of creation: 2006-08
% Last change:     2006-08-25
```

Uses connectionHyp.m.

3.5 script_connectionHyp.m

```
% script to run statistics using TWstat
% Author:          Florian Nisbach
% Date of creation: 2006-08
% Last change:     2006-08-18
```

This was used to generate the main dataset.

4 Generated Datasets

4.1 data_connectionHyp2006_08_26.mat

Dataset generated by script_connectionHyp.m.

4.2 quantiles_data_2006_09_04.mat

Now obsolete dataset generated by degree_script.m, because degrees.m was still faulty then.

4.3 quantiles_data_2006_09_18.mat

Corrected version of the above.

4.4 SW_08-15-dataset.mat

Old version of the small world factors arrays. Small parameter set, and some errors.

4.5 SW_08-18-dataset.mat

New version of the above.

4.6 timewin_data_08_15.mat

Old version of the main data set with small parameter set.

4.7 timewin_data_08_18.mat

New version of the above, with larger parameter set.