Noname manuscript No.
(will be inserted by the editor)

Learning from Complex networks: A Journey Towards Non-Scale-Freeness

Tanujit Chakraborty · Swarup Chattopadhyay · Suchismita Das · Ali Kashif Bashir

Received: date / Accepted: date

SOURCE CODE AND SUPPLEMENTARY MATERIALS

1 R Source Code for probability function, likelihood function of the proposed family of GLM distributions

```
rm(list = ls())
library(optimx)
library(Metrics)
library(MASS)
library(CORElearn)
```

"***Input Data***"

data←read.csv('data_degree_frequency.csv')

y1=data$C1

freq1←data$C2

fulldata←numeric(max(y1))

for(k in 1:max(y1))

{

if(is.element(k,y1)==TRUE)

{

index=match(k,y1)

fulldata[k]=freq1[index]

}

else

Address(es) of author(s) should be given

```
fulldata[k]=0
}
y←1:max(y1)
freq←fulldata
var1←sum(freq)
"Calculation of CV"
total_data_pts=sum(freq)
mean1=(t(freq)%*%y)/total_data_pts
sd1=sqrt((t(freq)%*%(y-mean1)^2)/total_data_pts)
print(mean1) print(sd1)
data_set_total=rep(y,freq)
mean2=mean(data_set_total)
sd2=sd(data_set_total)
print(mean2) print(sd2)
CV1=sd1/mean1
CV2=sd2/mean2
print(CV1) print(CV2)
```

GLM Type-I Distribution

```
"Likelihood function"
GLM_TypeI.lik←function(vector1,freq)
{
alpha=vector1[1]
beta=vector1[2]
gama=vector1[3]
n←sum(freq)
total_value=0
large_N=max(degree)
for(i in 1 large_N)
{
```

term1=alpha∗((1+i/gama)$^{(-alpha-1)}$)

term2=(1+(beta/((1+log(1+i/gama))^2)))

term3=exp(-alpha∗beta∗(log(1+i/gama)/(1+log(1+i/gama))))

total_value = total_value + term1∗term2∗term3

print("———")

}

C=1/total_value

print(C)

likterm1=n∗log(C)

likterm2=n∗log(alpha)

likterm3=(alpha+1)∗(t(freq)% ∗ %(log(1+y/gama)))

likterm4=t(freq)% ∗ %(log(1+(beta/(1+(log(1+y/gama)))^2)))

likterm5=alpha∗beta∗(t(freq)% ∗ %(log(1+y/gama)/(1+log(1+y/gama))))

final_likterm←sum(c(likterm1,likterm2,-likterm3,likterm4,-likterm5))

return(-final_likterm)

}

output11←optim(c(1,0,1),GLM_TypeI.lik, freq = freq)

print(output11)

"***Probability Function***"

probability.fun_GLM_TypeI ← function(vector3,datafile)

{

estalpha=vector3[1]

estbeta=vector3[2]

estgama=vector3[3]

x=datafile

sum2=0

large_N=max(degree)

for(i in 1 large_N)

{

pdf_term1=estalpha∗((1+i/estgama)$^{(-estalpha-1)}$)

pdf_term2=(1+(estbeta/((1+log(1+i/estgama))^2)))

pdf_term3=exp(-estalpha∗estbeta∗(log(1+i/estgama)/(1+log(1+i/estgama))))

sum2 = sum2 + pdf_term1∗pdf_term2∗pdf_term3

print("———")

}

C2=1/sum2

print(C2)

pdf_term4=estalpha∗((1+x/estgama)$^{(-estalpha-1)}$)

pdf_term5=(1+(estbeta/((1+log(1+x/estgama))^2)))

pdf_term6=exp(-estalpha∗estbeta∗(log(1+x/estgama)/(1+log(1+x/estgama))))

fina1_pdf_term=C2∗pdf_term4∗pdf_term5∗pdf_term6

return(fina1_pdf_term)

}

outfun←probability.fun_GLM_TypeI(c(output11$par[1],output11$par[2],output11$par[3]),y)

plot(y,outfun)

finaloutput←outfun∗var1

write.csv(finaloutput,'output_GLM_TypeI.csv')

GLM Type-II Distribution

"***Likelihood function***"

GLM_TypeII.lik←function(vector1,freq)

{

alpha=vector1[1]

beta=vector1[2]

gama=vector1[3]

n←sum(freq)

total_value=0

large_N=max(degree)

for(i in 1 large_N)

{

term1=(1+(beta/(1+(i/gama))))

term2=(alpha/(i+gama))

term3=(1+(i/gama))$^{(-alpha)}$

term4=exp(-alpha∗beta∗((i/gama)/(1+(i/gama))))

total_value = total_value + term1∗term2∗term3∗term4

print("———")

}

C=1/total_value

print(C)

likterm1=n∗log(C)

likterm2=t(freq)% ∗ %(log((gama+y+beta∗gama)/(gama+y)))

likterm3=t(freq)% ∗ %(log((alpha)/(gama+y)))

likterm4=alpha∗(t(freq)% ∗ %(log(1+(y/gama))))

likterm5=alpha∗beta∗(t(freq)% ∗ %(y/(gama+y)))

final_likterm←sum(c(likterm1,likterm2,likterm3,-likterm4,-likterm5))

return(-final_likterm)

}

output11←optim(c(1,0,1),GLM_TypeII.lik, freq = freq)

print(output11)

"***Probability Function***"

probability.fun_GLM_TypeII ← function(vector3,datafile)

{

estalpha=vector3[1]

estbeta=vector3[2]

estgama=vector3[3]

x=datafile

sum2=0

large_N=max(degree)

for(i in 1 large_N)

{

pdf_term1=(1+(estbeta/(1+(i/estgama))))

pdf_term2=(estalpha/((i+estgama)))

pdf_term3=(1+(i/estgama))$^{(-estalpha)}$

pdf_term4=exp(-estalpha∗estbeta∗((i/estgama)/(1+(i/estgama))))

sum2 = sum2 + pdf_term1∗pdf_term2∗pdf_term3∗pdf_term4

print("———")

}

C2=1/sum2

print(C2)

pdf_term5=(1+(estbeta/(1+(x/estgama))))

pdf_term6=(estalpha/((x+estgama)))

pdf_term7=(1+(x/estgama))$^{(-estalpha)}$

pdf_term8=exp(-estalpha∗estbeta∗((x/estgama)/(1+(x/estgama))))

fina1_pdf_term=C2∗pdf_term5∗pdf_term6∗pdf_term7∗pdf_term8

return(fina1_pdf_term)

}

outfun←probability.fun_GLM_TypeII(c(output11$par[1],output11$par[2],output11$par[3]),y)

plot(y,outfun)

finaloutput←outfun∗var1

write.csv(finaloutput,'output_GLM_TypeII.csv')

GLM Type-III Distribution

"***Likelihood function***"

GLM_TypeIII.lik ← function(vector1,freq)

{

alpha=vector1[1]

beta=vector1[2]

gama=vector1[3]

n ← sum(freq)

total_value=0

large_N=max(degree)

for(i in 1 large_N)

{

term1=1+((beta∗(log(1+i/gama)))/(i/gama))

term2=((i/gama)/(1+i/gama))$^{(beta)}$

term3=(alpha/(i+gama))

term4=exp(-alpha∗log((1+i/gama)∗(((i/gama)/(1+i/gama))^{beta})))

total_value = total_value + term1∗term2∗term3∗term4

print("———")

}

C=1/total_value

print(C)

likterm1=n∗log(C)

likterm2=t(freq)% ∗ %(log((y/gama + beta∗log(1+y/gama))/(y/gama)))

likterm3=beta∗(t(freq)% ∗ %(log((y/gama)/(1+y/gama))))

likterm4=t(freq)% ∗ %(log(alpha/(y+gama)))

likterm5=alpha∗(t(freq)% ∗ %(log((1+y/gama)∗(((y/gama)/(1+y/gama))$^{(beta)}$))))

final_likterm ← sum(c(likterm1,likterm2,likterm3,likterm4,-likterm5))

return(-final_likterm)

}

output11 ← optim(c(1,0,1),GLM_TypeIII.lik, freq = freq)

print(output11)

"***Probability Function***"

probability.fun_GLM_TypeIII ← function(vector3,datafile)

{

estalpha=vector3[1]

estbeta=vector3[2]

estgama=vector3[3]

x=datafile

sum2=0

```
large_N=max(degree)

for(i in 1 large_N)

{

pdf_term1=1+((estbeta*(log(1+i/estgama)))/(i/estgama))

pdf_term2=((i/estgama)/(1+i/estgama))^(estbeta)

pdf_term3=(estalpha/(i+estgama))

pdf_term4=exp(-estalpha*log((1+i/estgama)*(((i/estgama)/(1+i/estgama))^estbeta)))

sum2 = sum2 + pdf_term1*pdf_term2*pdf_term3*pdf_term4

print("———")

}

C2=1/sum2

print(C2)

pdf_term5=1+((estbeta*(log(1+x/estgama)))/(x/estgama))

pdf_term6=((x/estgama)/(1+x/estgama))^(estbeta)
pdf_term7=(estalpha/(x+estgama))

pdf_term8=exp(-estalpha*log((1+x/estgama)*(((x/estgama)/(1+x/estgama))^estbeta)))

fina1_pdf_term=C2*pdf_term5*pdf_term6*pdf_term7*pdf_term8

return(fina1_pdf_term)

}

outfun←probability.fun_GLM_TypeIII(c(output11$par[1],output11$par[2],output11$par[3]),y)

plot(y,outfun)

finaloutput←outfun*var1

write.csv(finaloutput,'output_GLM_TypeIII.csv')
```

GLM Type-IV Distribution

```
"Likelihood function"

GLM_TypeIV.lik←function(vector1,freq)

{

alpha=vector1[1]

beta=vector1[2]
```

```
    sigma=vector1[3]

    n←sum(freq)

    total_value=0

    large_N=max(degree)

    for(i in 1:large_N)

    {

    term1=alpha/sigma

    term2=(log((i/sigma)+1)+1+beta)/((i/sigma)+1)

    term3=((log((i/sigma)+1))^(beta))/((log((i/sigma)+1)+1)^(beta+1))

    term4=exp(-alpha.(((log((i/sigma)+1))^(beta+1))/((log((i/sigma)+1)+1)^(beta))))

    total_value = total_value + term1*term2*term3*term4

    }

    C=1/total_value

    likterm1=n*log(C)

    likterm2=n*log(alpha)

    likterm3=t(freq)% * %log(y+sigma)

    likterm4=t(freq)% * %(log(log((y/sigma)+1)+1+beta))

    likterm5=beta*(t(freq)% * %(log(log((y/sigma)+1))))

    likterm6=(beta+1)*(t(freq)% * %(log(log((y/sigma)+1)+1)))

    likterm7=alpha*(t(freq)% * %(((log((y/sigma)+1))^(beta+1))/((log((y/sigma)+1)+1)^(beta))))

    final_likterm ← sum(c(likterm1,likterm2,-likterm3,likterm4,likterm5,-likterm6,-likterm7))

    return(-(final_likterm))
}

    output11 ← optim(c(1,0,1),GLM_TypeIV.lik, freq = freq)

    print(output11)

    "Probability Function"

    probability.fun_GLM_TypeIV←function(vector3,datafile)
{

    estalpha=vector3[1]

    estbeta=vector3[2]

    estsigma=vector3[3]
```

```
x=datafile

sum2=0

large_N=max(degree)

for(i in 1:large_N)


pdf_term1=estalpha/estsigma

pdf_term2=(log((i/estsigma) +1)+1+estbeta)/((i/estsigma)+1)

pdf_term3=((log((i/estsigma)+1))^(estbeta))/((log((i/estsigma)+1)+1)^(estbeta+1))

pdf_term4=exp(-estalpha*(((log((i/estsigma)+1))^(estbeta+1))/((log((i/estsigma)+1)+1)^(estbeta))))

sum2 = sum2 + pdf_term1*pdf_term2*pdf_term3*pdf_term4

print("———")

}

print('
C2=1/sum2

print(sum2)

print(C2)

pdf_term5=estalpha/estsigma

pdf_term6=(log((x/estsigma)+1)+1+estbeta)/((x/estsigma)+1)

pdf_term7=((log((x/estsigma)+1))^(estbeta))/((log((x/estsigma)+1)+1)^(estbeta+1))

pdf_term8=exp(-estalpha*(((log((x/estsigma)+1))^(estbeta+1))/((log((x/estsigma)+1)+1)^(estbeta))))

fina1_pdf_term=C2*pdf_term5*pdf_term6*pdf_term7*pdf_term8

return(fina1_pdf_term)

}

outfun←probability.fun_GLM_TypeIV(c(output11$par[1],output11$par[2],output11$par[3]),y)

plot(y,outfun)

finaloutput←outfun.var1

print(finaloutput)

write.csv(finaloutput,'output_GLM_TypeIV.csv')
```

Calculation of test statistics

```
data1←read.csv('output_GLM_TypeI / II / III / IV.csv')
y1=data1

actual_freq←data1.Actual

estimated_freq←data1.GLM_TypeI / II / III / IV

actual_chisquare_value←sum(((actual_freq-estimated_freq)^2)/estimated_freq)

array_of_synthetic_chisquare_value←rep(0,50000)

for(i in 1:50000)

{

print(i)

synthetic_data=sample(1:max(y1),sum(actual-freq),prob=actual-freq,rep=T)

frequ_table_synthetic_data←as.data.frame(table(synthetic_data))

unique_degree←as.numeric(as.character(frequ_table_synthetic_data.synthetic_data))

unique_freq←as.numeric(as.character(frequ_table_synthetic_data.Freq))

y←1:max(y1)

synthetic_freq←rep(0,max(y1))

synthetic_freq[unique_degree]←unique_freq

synthetic_chisquare_value←sum(((synthetic_freq-estimated_freq)^2)/estimated_freq)

array_of_synthetic_chisquare_value[i]←synthetic_chisquare_value

}

hist(array_of_synthetic_chisquare_value)

p_value= mean(array_of_synthetic_chisquare_value>actual_chisquare_value)

print(p_value)

print('RMSE')

print(rmse(actual_freq,estimated_freq))

print('MAE')

print(mae(actual_freq,estimated_freq))

kldivergence←KL.plugin(actual_freq,estimated_freq)

print('kldivergence')

print(kldivergence)
```

Matlab Source Code for plotting degree frequency

```
clear all;
data=csvread('input_datra.csv',1);
xDeg=data(:,1);
print('Unique degree ')
xAct=data(:,2);
print('Actual Frequency')
xGLM_TypeI=data(:,3);
print('GLM_TypeI Frequency')
xGLM_TypeII=data(:,4);
print('GLM_TypeII Frequency')
xGLM_TypeIII=data(:,5);
print('GLM_TypeIII Frequency')
xGLM_TypeIV=data(:,6);
print('GLM_TypeIV Frequency')
xLomax=data(:,7);
print('Lomax Frequency')
xPow=data(:,8);
print('Power-law Frequency')
xPar=data(:,9);
print('Pareto Frequency')
xLog=data(:,10);
print('Log-normal Frequency')
xPoC=data(:,11);
print('Power-law with Cutoff Frequency')
xExp=data(:,12);
print('Exponential Frequency')
figure
loglog(xDeg,xAct,'.','MarkerSize',7,'MarkerEdgeColor','b')
```

```
hold on loglog(xDeg,xPar,'linewidth',1,'color',[0, 0.75, 0.75])

hold on loglog(xDeg,xPow,'linewidth',1,'color',[0.4940, 0.1840, 0.5560])

hold on loglog(xDeg,xLog,'linewidth',1,'color',[0.75,0,0.75])

hold on loglog(xDeg,xPoC,'linewidth',1,'color',[0.75,0.75,0])

hold on loglog(xDeg,xExp,'linewidth',1,'color',[0.25,0.25,0.25])

hold on loglog(xDeg,xLomax,'linewidth',1,'color',[0, 0.4470, 0.7410])

hold on loglog(xDeg,xGLM_TypeI,'linewidth',1.2,'color',[0.8500,0.3250, 0.0980])

hold on loglog(xDeg,xGLM_TypeII,'linewidth',1.4,'color',[0.9290, 0.6940, 0.1250])

hold on loglog(xDeg,xGLM_TypeIII,'linewidth',1.6,'color',[0, 0.5, 0])

hold on loglog(xDeg,xGLM_TypeIV,'linewidth',1.8,'color',[1, 0, 0])

ylim([0.3 1000000]);

xlim([1 100000]);

set(gca,'fontweight','bold','fontsize',12); xlabel('Node Degree'); ylabel('Frequency');

L=legend('Input Network','Pareto Type-I','Power law','Log-normal','Power law
cutoff','Exponential','Lomax','GLM_TypeI','GLM_TypeII','GLM_TypeIII','GLM_TypeIV','Location',[0.5, 0.5,
.25, .25]);
```

2 Description of datasets

The data sets we study here come from variety of different disciplines. We present results of fitting double power-law distribution over 50 real world complex networks which are available online [1, 2]: Large online social networks (Social circles from Twitter (eg–Twitter), Social circles from Google+ (ego-Gplus), Salshdot social network (soc-Salshdot), Delicious online social network (soc-Delicious), Digg online social network (soc-Digg), Academia online social network (soc-Academia), Live Journal online social network (Live-Journal), Dogster friendship networks (soc-Dogster), Spreading processes of the announcement of the discovery of a new particle with the features of the Higgs boson on 4th July 2012 (Higgs-Twitter), Gemsec Facebook dataset (Artist-Facebook network, Athletes-Facebook network)), citation networks (Arxiv High Energy Physics paper citation network (cit-HepPh), Arxiv High Energy Physics Theory citation network (cit-HepTh), Citation network among US Patents (cit-Patents), citation network extracted from the CiteSeer digital library (cit-Citeseer)), collaboration networks of co-authorships from DBLP and various areas of physics (Collaboration network of Arxiv Astro Physics (ca-AstroPh), Collaboration network of Arxiv Condensed Matter (ca-CondMat), Collaboration network of Arxiv General Relativity (ca-GrQc), Collaboration network of Arxiv High Energy Physics (ca-HepPh), Collaboration network of Arxiv High Energy Physics Theory (ca-HepTh)), web and blog graphs (Web Graph from Google (Google), Web graph of Berkeley and Stanford (BerkStan), Web graph of Wikipedia on 2009 (Wikipdia2009), Web graph of Wikipedia Link Fr (WikipediaLinkFr), A directed network of hyperlinks between the articles of the Chinese online encyclopedia Hudong (Web-Hudong)), Biological Networks (Protein protein interaction network in budding yeast (Yeast-PPIN), Mouse gene regulatory network (Bio-Mouse-Gene) and a network of disorders and disease genes (Diseasome), protein-protein interactions (Bio-Dmela), Gene functional associations network (Bio-WormNet-v3)), product co-purchasing networks (Amazon product co-purchasing network from March12 2003 (amazon0312), Amazon product co-purchasing network from May5 2003 (amazon0505), Amazon product co-purchasing network from June1 2003 (amazon0601)), Temporal networks (Comments, questions, and answers on Math Overflow (sx-mathoverflow), Comments, questions, and answers on Stack Overflow (sx-stackoverflow), Comments, questions, and answers on Super User (sx-superuser), Comments, questions, and answers on Ask Ubuntu (sx-askubuntu)), Communication networks (Email Communication

network from Enron (Email-Enron), Wikipedia talk network (Wiki-Talk), Network is from a Czech dating site (Rec-Libimseti)), Networks with ground-truth communities (Network of Wikipedia hyperlinks (com-Wiki-Topcats), Friendster online social network (com-Frienster), LiveJournal online social network (com-LiveJournal), Orkut online social network (com-Orkut), Youtube online social network (com-Youtube)) and Brain networks(Edges represent fiber tracts that connect one vertex to another (bn-human-BNU-1-0025890-session-1, bn-human-BNU-1-0025890-session-2, bn-human-BNU-1-0025864-session-2, bn-human-BNU-1-0025913-session-2, bn-human-BNU-1-0025886-session-1)).

References

1. Leskovec J, Krevl A (2014) SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data
2. Rossi RA, Ahmed NK (2015) The network data repository with interactive graph analytics and visualization. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, URL http://networkrepository.com