Using the Marshall-Olkin Extended Zipf Distribution in Graph Generation

Ariel Duarte-López[1(✉)], Arnau Prat-Pérez[1], and Marta Pérez-Casany[2]

[1] DAMA-UPC, Departament d'Arquitectura de Computadors, Universitat Politècnica de Catalunya, Barcelona, Spain
{aduarte,aprat}@ac.upc.edu
[2] DAMA-UPC, Departament Matemàtica Aplicada II, Universitat Politècnica de Catalunya, Barcelona, Spain
marta.perez@upc.edu

Abstract. Being able to generate large synthetic graphs resembling those found in the real world, is of high importance for the design of new graph algorithms and benchmarks. In this paper, we first compare several probability models in terms of goodness-of-fit, when used to model the degree distribution of real graphs. Second, after confirming that the MOEZipf model is the one that gives better fits, we present a method to generate MOEZipf distributions. The method is shown to work well in practice when implemented in a scalable synthetic graph generator.

1 Introduction

The analysis of large real graphs has attracted the interest of the industry and academia due to its multiple applications, and as a consequence, many technologies for their analysis have emerged. In order to fairly compare the performance and features of such technologies, several benchmarking initiatives have kicked off [2,4,6]. In general, these initiatives use synthetic graph generators in seek for the flexibility not always found in real datasets.

Being able to generate credible graphs that mimic the characteristics of the real ones is of high importance, because they directly impact the performance of the systems under test. One of these characteristics is the degree distribution of the nodes in the graph. In general, it is widely accepted that in real networks the degree sequence follows a power-law, since the majority of the nodes have a small degree while just few of them are connected to many neighbours, thus having a very large degree [7].

A particular power law distribution with support the strictly positive integer numbers is the Zipf distribution. The Zipf's law shows a linear shape in the log-log scale, but in practice this is not always the case in real networks, where it is only observed for degree values large enough. For low degree nodes, the plot usually shows concavity, and less often, convexity [3]. One generalization of the Zipf's law that solves this issue is the Marshall-Olkin extended Zipf distribution (MOEZipf), which uses the Marshall-Olkin transformation to add an extra parameter that gives more flexibility to the family.

S. Hunold et al. (Eds.): Euro-Par 2015 Workshops, LNCS 9523, pp. 493–502, 2015.
DOI: 10.1007/978-3-319-27308-2_40

In this paper, we first prove by means of an analysis of several real graphs, the suitability of the MOEZipf model as a degree distribution. Second, we propose a method to generate degree sequences following the MOEZipf distribution, which is implemented in a scalable graph generator (The LDBC Data Generator [4]), showing that it works well in practice.

The paper is structured as follows: In Sect. 2, we introduce the MOEZipf probability distribution. In Sect. 3 we show that MOEZipf adjusts well the distributions of real graphs. In Sect. 4, we propose a method to generate random samples from a MOEZipf distribution. In Sect. 5, we show the results obtained with Datagen using the proposed approach, in Sect. 6, we conclude the paper.

2 The MOEZipf Model

A random variable (r.v.) X is said to follow a Zipf distribution with *scale* parameter $\alpha > 1$ if, and only if, its probability mass function (pmf) is equal to:

$$P(X = x) = \frac{x^{-\alpha}}{\xi(\alpha)}, \ for \ \ x = 1, 2, 3, \cdots, \tag{1}$$

where $\xi(\alpha) = \sum_{k=1}^{+\infty} k^{-\alpha}$ is the Riemann zeta function. The Zipf distribution is often suitable to fit data that correspond to frequencies of frequencies or to ranked data. This type of data shows a widespread pattern in their measurements with a very large probability at one and a very small probability at some very large values. Taking logarithms at (1), one obtains that the Zipf distribution shows a linear pattern in the log-log scale since

$$\log(P(X = k)) = -\alpha \log(x) - \log(\xi(\alpha)).$$

This linearity is useful to check whether the data may be well fitted or not by means of the Zipf distribution, by just plotting the empirical probabilities. However, in practice usually this linearity is just observed in the tail of the distribution, while a concavity is observed at the beginning. The MOEZipf distribution is proposed in [8], as an approximate model to adapt this behavior.

A r.v. X is said to follow a MOEZipf distribution with parameters α and β if, and only if, its survival function (SF) is equal to:

$$P(X > x) = \overline{G}(x; \alpha, \beta) = \frac{\beta \ \overline{F}(X)}{1 - \overline{\beta}\,\overline{F}(X)} = \frac{\beta \ \xi(\alpha, x+1)}{\xi(\alpha) - \overline{\beta}\,\xi(\alpha+1)}, \tag{2}$$

for $\beta > 0$, $\alpha > 1$ and $\overline{\beta} = 1 - \beta$. Being $\overline{F}(x)$ the SF of the Zipf(α) distribution.

The pmf of the MOEZipf may be computed by means of

$$\begin{aligned} P(X = x) &= \overline{G}(x-1; \alpha, \beta) - \overline{G}(x; \alpha, \beta) \\ &= \frac{x^{-\alpha}\,\beta\,\xi(\alpha)}{[\xi(\alpha) - \overline{\beta}\xi(\alpha, x)][\xi(\alpha) - \overline{\beta}\xi(\alpha, x+1)]}, \quad x = 1, 2, 3, \cdots, \end{aligned} \tag{3}$$

Table 1. Main characteristics of the nine real networks analysed.

Network	Nodes	Edges	GCC	ACC	AD	Type
Amazon	$262K$	$1.24M$	0.2361	0.4198	0.0027	Directed
CA roads	$1.97M$	$5.53M$	0.0604	0.0464	0.1260	Undirected
DBLP	$317K$	$105M$	0.3064	0.6324	0.2665	Undirected
Livejournal	$4M$	$34.68M$	0.1253	0.2843	0.045	Undirected
NotreDame	$326K$	$1.5M$	0.0877	0.2346	−0.0617	Directed
Patents	$3.78M$	$16.52M$	0.0671	0.0757	0.1332	Directed
TX roads	$1.38M$	$3.84M$	0.0602	0.0470	0.1304	Undirected
Wikipedia	$2.39M$	$5.02M$	0.0022	0.0526	−0.0853	Directed
Youtube	$1.14M$	$2.99M$	0.0062	0.0808	−0.0369	Undirected

where $\xi(\alpha, x) = \sum_{k=x+1}^{+\infty} k^{-\alpha}$ is the Hurwitz Zeta function with parameter α. When $\beta = 1$, in (3) one obtains the pmf of the Zipf(α) distribution. An advantage of the MOEZipf distribution is that it shows a convexity or concavity behaviour at the beginning of the distribution depending on whether $0 < \beta < 1$ or $\beta > 1$ respectively, while keeping the linearity in the tail.

3 Real Graphs Analysis

This paper is motivated after the analysis of the degree distribution of nine real networks coming from diverse domains[1], using eight different probabilistic models: Geometric, Poisson, Zipf, Right-truncated Zipf, Altmann, MOEZipf, Negative Binomial and Discrete Weibull. Table 1 shows the number of nodes, number of edges, global clustering coefficient (GCC), average clustering coefficient (ACC), assortativity degree (AD) and directionality of the networks analysed. For each directed networks, both the in-degree (In) and out-degree (Out) sequences were analysed, making a total of 13 degree sequences.

The Zipf, the Right-truncated Zipf and the MOEZipf probability distributions have been considered mainly because of two reasons. On one side, because the Zipf distribution is assumed to be the node degree distribution in most scientific papers, and its Right-truncated version is a way to improve the fit in the tail of the distribution. On the other side, because we are interested in proving the suitability of the Zipf generalization: the MOEZipf distribution.

The Poisson and the Negative Binomial distributions have been included for being the first the classical distribution for counts when the events take place randomly and with the same probability, and the second its usual biparametric alternative used when the data show more dispersion than it was initially expected. However, we have observed this is not our case, since fitting the Negative Binomial oftenly results in numerical problems and when not, the fits are not satisfactory.

[1] Networks downloaded from http://snap.stanford.edu/datarepository.

The reason for including the Geometric and the Discrete Weibull distributions is clearly different. These distributions may be seen, respectively, as the discrete versions of the Exponential and the Weibull, which are the continuous distributions associated to *time to an event* r.v.. One advantage of the Geometric is its simplicity and that it does not require the truncation at one, because its support are the strictly positive integer numbers. The Discrete Weibull is useful when the lifetime is measured counting cycles, shocks or revolutions. In our case, it has sense think about that an individual being *active* or *alive* while he is able to create connections with the others. From this point of view, the distribution comes naturally if one thinks that the lifetime is measured counting the number of connections performed. Finally, the Altmann distribution, also known as Zipf-Alekseev distribution or Zipf with an exponential cuttoff, is used in quantitative linguistics and it is also a bi-parameter generalization of the Zipf. In this case, it is assumed that the support is finite and the tail decreases quickly since the probabilities are multiplied by $e^{-x\beta}$.

In order to fit the degree sequence for a given graph, the maximum likelihood parameter estimations were calculated by means of the *mle* function included in the R software [9]. It is known that maximum likelihood parameter estimations are good, because they are unbiased and have minimum variance. The models were compared using the *Akaike Information Criterion* (AIC) and the *Bayesian Information Criterion* (BIC) goodness of fit measures [1], which are defined as:

$$AIC = -2l(\hat{\theta}, k) + 2M\frac{N}{N - M - 1}$$

and

$$BIC = -2l(\hat{\theta}, k) + Mlog(N)$$

respectively, where $l(\hat{\theta}, k)$ is the value of the log-likelihood function evaluated at $\hat{\theta}$, the maximum likelihood estimation of θ, for a given degree sequence k. M is the number of parameters of each probabilistic model (in our case it is equal to one or two) and N is the number of nodes of the network.

Table 2 shows the ΔAIC and ΔBIC for each network and all the models. These values were computed by means of the difference between the value in the current model and the value in the best model. Therefore, for each network the best model is the one that has a zero value in ΔAIC and ΔBIC.

Our experiments reveal that the analysed degree sequences can be explained with just three out of the eight models considered, which are: the MOEZipf, the Discrete Weibull and the Altmann models. The MOEZipf model is the best in 54 % of the cases, followed by the Discrete Weibull in 38 % of the cases and the Altmann in 8 % of the cases.

Figure 1 shows four degree sequences associated to the networks Amazon (In), DBLP, Patents (Out) and Youtube respectively; jointly with the fit of the best four models in each case. In all the cases the plots are in log-log scale. The best fit for the Amazon (In) is given by the MOEZipf model with parameter estimations $\hat{\alpha} = 3.0295$ and $\hat{\beta} = 27.1284$, the second best model in this case is the Discrete

Table 2. Values of the ΔAIC and ΔBIC for the different networks and probabilistic models.

Networks		Geometric	Poisson	Zipf	Right-trun. Zipf	Altmann	MOEZipf	Neg. Bin.	Discrete Weibull
Amazon (In)	ΔAIC	10772.3325	525546.2495	180203.9906	166775.5608	10774.3423	**0**	10768.3804	9774.4649
	ΔBIC	10761.856	525535.773	180193.5141	166775.5608	10774.3423	**0**	10768.3804	9774.4649
Amazon (Out)	ΔAIC	636386.6206	290672.7867	1060794.3393	593382.9374	636389.4636	**0**	–	–
	ΔBIC	636376.1615	290662.3277	1060783.8803	593382.9373	636389.4635	**0**	–	–
CA roads	ΔAIC	1738132.797	1118395.2451	3696976.2046	2703401.834	1738138.1355	519401.402	–	**0**
	ΔBIC	1738120.3059	1118382.754	3696963.7135	2703401.834	1738138.1355	519401.402	–	**0**
DBLP	ΔAIC	68821.822	1256991.2234	36537.236	28493.1692	2781.7322	2526.152	–	**0**
	ΔBIC	68811.672	1256981.0734	36527.086	28493.1692	2781.7322	2526.1521	–	**0**
Livejournal	ΔAIC	1472010.06	66043193.8352	744846.7078	691016.3788	59123.1575	86329.1964	58895.0102	**0**
	ΔBIC	1471997.4005	66043181.1758	744834.0483	691016.3787	59123.1574	86329.1964	58895.0101	**0**
NotreDame (In)	ΔAIC	486831.1503	5007706.9131	74.3956	34.4528	–	**0**	–	–
	ΔBIC	486820.4566	5007696.2193	63.7018	34.4528	–	**0**	–	–
NotreDame (Out)	ΔAIC	71534.3133	2643273.0597	72029.2877	66891.3556	23274.8034	**0**	22562.4914	10614.9758
	ΔBIC	71524.4788	2643263.2252	72019.4531	66891.3556	23274.8034	**0**	22562.4914	10614.9758
Patents (In)	ΔAIC	385629.9996	12328277.4843	1116962.9351	1030887.7683	32525.9313	99456.9783	28490.1532	**0**
	ΔBIC	385617.0027	12328264.4875	1116949.9382	1030887.7683	32525.9313	99456.9784	28490.1532	**0**
Patents (Out)	ΔAIC	275808.2025	7055231.2285	2608735.5631	2368252.8935	275811.9408	**0**	221032.981	–
	ΔBIC	275795.6501	7055218.6762	2608723.0108	2368252.8935	275811.9408	**0**	221032.981	–
TX roads	ΔAIC	1160225.2518	793426.0519	2497743.0319	1827262.7551	1160229.4974	394526.2436	–	**0**
	ΔBIC	1160213.1142	793413.9144	2497730.8943	1827262.7551	1160229.4973	394526.2436	–	**0**
Wikipedia (In)	ΔAIC	2205641.8423	13174755.7601	162.2935	153.6566	–	**0**	–	–
	ΔBIC	2205629.1642	13174743.082	149.6155	153.6567	–	**0**	–	–
Wikipedia (Out)	ΔAIC	651885.7283	32209782.8763	57.4127	5.3241	**0**	59.4128	–	–
	ΔBIC	651875.8262	32209772.9742	47.5106	5.3241	**0**	59.4129	–	–
Youtube	ΔAIC	581583.7338	11558597.1877	20322.1168	20032.0998	13402.6611	**0**	–	1939.5714
	ΔBIC	581572.8997	11558586.3536	20311.2827	20032.0998	13402.6611	**0**	–	1939.5714

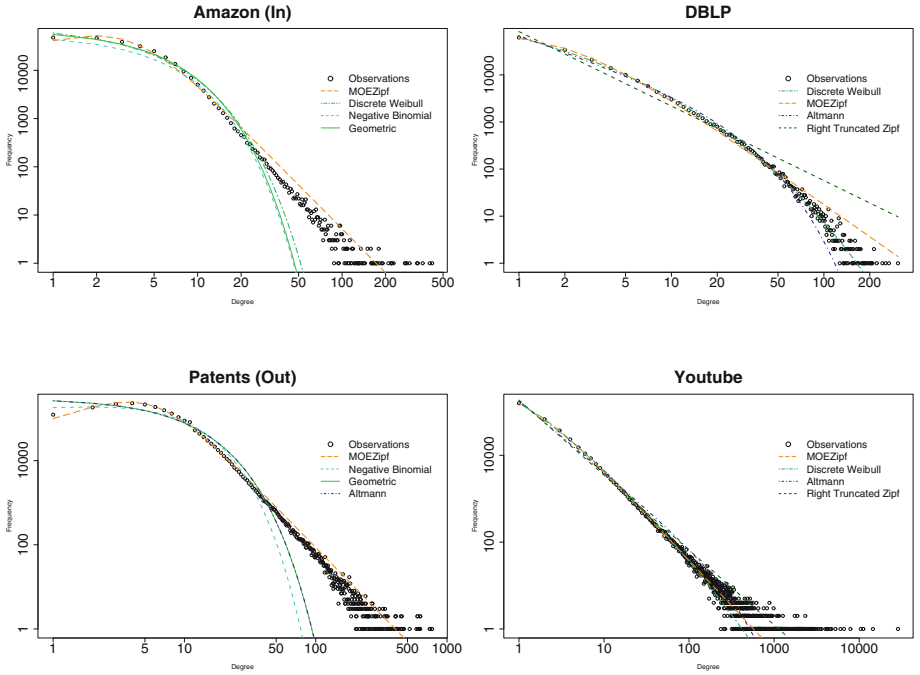

Fig. 1. Observed degree sequences in the Amazon (In), DBLP, Patents (Out) and Youtube networks jointly with the four best models in each case.

Weibull with parameters $\hat{p} = 0.7519$ and $\hat{\beta} = 0.9271$. The model that gives the best fit to the DBLP degree sequence is the Discrete Weibull with parameter estimations $\hat{p} = 0.2622$ and $\hat{\beta} = 0.3881$, followed by the MOEZipf model with parameters $\hat{\alpha} = 2.2767$ and $\hat{\beta} = 4.8613$. For the Patents (Out) degree sequence, the best model is the MOEZipf with parameters $\hat{\alpha} = 3.196$ and $\hat{\beta} = 119.264$ and, in second place, the Negative Binomial model with parameters $\hat{\gamma} = 1.4873$ and $\hat{p} = 0.8317$. The best model for the Youtube network is the MOEZipf with parameters $\hat{\alpha} = 2.089$ and $\hat{\beta} = 2.4101$, and the second best model is the Discrete Weibull with parameters $\hat{p} = 0.0044$ and $\hat{\beta} = 0.1424$. The information about how well a model behaves with respect to the others can be found in Table 2.

4 Generating MOEZipf Degree Samples

The proposed method for generating MOEZipf degree sequences is based on the well known *Inverse Principle* [5]. Given a sequence of uniformly distributed random values between 0 and 1, we obtain a sequence of values of the target distribution using its inverse cumulative probability function (cpf). Given that, the cpf is equal to one minus the SF, and that from (2) the SF of the MOEZipf is easily deduced from the SF of the Zipf, one can obtain the desired value by

applying the inverse principle to the Zipf distribution after properly modifying the generated uniform random value.

Algorithm 1 shows the pseudocode associated to this procedure. Given fixed values for α and β, we first initialize variable x to be equal to the first value in the support of the MOEZipf which is one. After generating a value u uniformly from 0 and 1, it is transformed to value u' as follows:

$$u' = \frac{u\beta}{1 + u(\beta - 1)}$$

If $1 - \frac{1}{u} \leq \beta$, the final x value is equal to the first integer value such that $u' \leq F_\alpha(x)$, where $F_\alpha(x)$ is the cpf of the Zipf distribution. Otherwise, the final x is equal to the first value satisfying $u' \geq F_\alpha(x)$.

Algorithm 1. MOEZipf Generator

```
1: procedure SAMPLE_MOEZIPF(α, β)
2:     x ← 1
3:     u ← uniform random number [0, 1]
4:     u' ← uβ / (1+u(β−1))
5:     loop
6:         z_c ← F_α(x)
7:         if β < 1  and  (u−1)/u >= β then
8:             if u' >= z_c then
9:                 return x
10:        else
11:            if u' <= z_c then
12:                return x
13:        x ← x + 1
```

5 Scalable MOEZipf Generation with Datagen

Datagen is the synthetic graph generator used in the LDBC Social Network Benchmark [4]. It is designed to generate social undirected networks with different degree distributions, with correlated attributes and edges connecting people with similar characteristics in an homophylic way. Datagen is implemented using the Map-Reduce parallel programming paradigm, and therefore is able to generate large graphs by running on small commodity clusters.

We have extended Datagen with the method proposed in Sect. 4 to generate MOEZipf based graphs scalably[2]. We have generated seven synthetic graphs with a degree distributions similar to those of the seven real degree sequences analysed in Sect. 3 where the MOEZipf distribution is the best fitting model: Amazon (In),

[2] The implemented plugin can be found at Datagen's source code repository https://github.com/ldbc/ldbc_snb_datagen.

Table 3. Parameters of the MOEZipf distribution estimated from the original networks vs the ones estimated from the networks generated using Datagen; the generation time of each network.

Networks	Original network estimated parameters		Synthetic network estimated parameters		Generation time (s)
	$\hat{\alpha}$	$\hat{\beta}$	$\tilde{\alpha}$	$\tilde{\beta}$	
Amazon (In)	3.0295	27.1284	3.0332	27.3464	991
Amazon (Out)	9.5281	6390058.5115	9.1074	3057506.2967	1451
NotreDame (In)	2.0174	1.0657	2.0259	1.0873	1598
NotreDame (Out)	2.4215	15.6546	2.4229	15.7044	1587
Patents (Out)	3.196	119.264	3.1959	119.2742	4600
Wikipedia (In)	2.5479	1.045	2.5457	1.0431	5505
Youtube	2.089	2.4101	2.0981	2.4534	2200

Amazon (Out), NotreDame (In), NotreDame (Out), Patents (Out), Wikipedia (In) and Youtube. To generate the graphs we have used the same number of nodes, and configured the implemented MOEZipf degree sequence generation with the same parameters as those estimated from the original networks. Note that Datagen only generates undirected graphs, but for the purpose of this paper, we are only interested in being able to mimic the degree distributions and to prove that these can be generated in a scalable way.

Table 3 shows, for each one of the networks the following information. On the one hand, the parameters $\hat{\alpha}$ and $\hat{\beta}$ estimated from the original networks, which are used to generate the synthetic ones. On the other hand, the parameters $\tilde{\alpha}$ and $\tilde{\beta}$ estimated from the resulting synthetic networks. We see that, for six out of seven cases, the resulting estimated parameters from the synthetic networks are very similar to those from the original graphs. Only for the Amazon (Out) degree sequence, there is a remarkable difference in the value of the β parameter. This is because the log-likelihood function, $l(\beta, \alpha; k)$, as a function of β tends to an asymptote as β increases. More exactly:

$$l(\beta, \alpha; k) \simeq N \log(\beta) + g(\alpha; k),$$

being $g(\alpha; k)$ a function that does not involve the β parameter. Thus, there are not significant differences between the values of the log-likelihood function for two β values if both are large enough. Finally, in the last column of Table 3 we see the time taken to generate these datasets in our test machine cluster, composed by four quad-core nodes with 32 GB of RAM each and 2 TB spinning disks. In general, we see that the generation model is able to accurately generate degree sequences with the desired characteristics, in a scalable way.

Figure 2 shows two examples of degree distributions of two synthetically generated graphs. Specifically, the ones generated to mimic the characteristics of the Patents (Out) and Youtube degree sequences. We also plot the theoretical

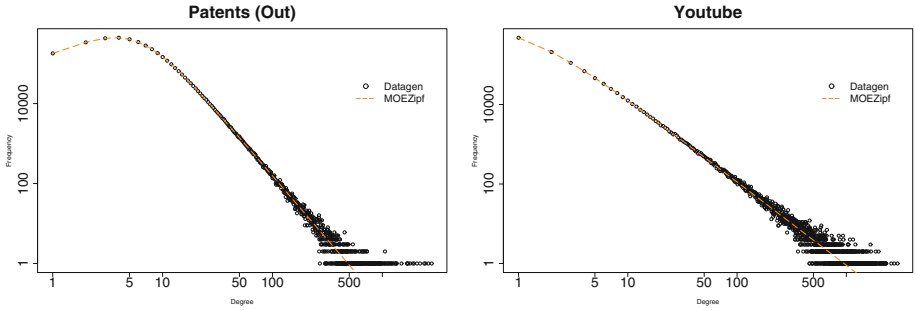

Fig. 2. Synthetically generated graphs with similar characteristics to the Patents (Out) (N = 3774767, $\hat{\alpha} = 3.196$ and $\hat{\beta} = 119.264$) and Youtube (N = 1134890, $\hat{\alpha} = 2.089$ and $\hat{\beta} = 2.4101$) graphs respectively.

MOEZipf degree distribution with the parameter estimations $(\tilde{\alpha}, \tilde{\beta})$. In both cases, we see that Datagen is able to generate a graph with a degree sequence with the same characteristics of the real ones accurately.

6 Conclusions and Future Work

We have analysed a set of degree distributions from real networks using several probability models. The (AIC) and BIC have been used to compare the different tested models. We have shown that the MOEZipf distribution is the one that better explains the degree distributions observed in real networks. Based on this result, we have presented a method to generate MOEZipf degree sequences, and implemented it as an extension to the LDBC graph generator, namely Datagen. Our experiments have shown that with the Datagen implementation, we can generate graphs with real degree distributions in a scalable way.

In this work, we have focused on generating realistic degree distributions. Future work will consist in developing techniques to reproduce other networks' structural characteristics, such as the clustering coefficient or the degree of assortativity. Moreover, currently Datagen only supports the generation of undirected graphs. In the future we will work on extending Datagen to generate directed graphs with different in-degree and out-degree distributions.

Acknowledgments. The authors, all members of DAMA-UPC, thank the Ministry of Science and Innovation of Spain, Generalitat de Catalunya, for grant numbers TIN2013-47008-R and SGR2014-890 respectively and also the EU FP7/2007-2013 for funding the LDBC project (ICT2011-8-317548). M. Pérez-Casany also thanks the Spanish Ministry of Education and Science for grant MTM2013-43992-R and Generalitat de Catalunya for grant 2014 SGR 890 (AGAUR). The authors thank Oracle Labs for the strategic support to the Graphalytics project.

References

1. Burnham, K.P., Anderson, D.R.: Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach. Springer-Verlag, New York (2002)
2. Capota, M., Hegeman, T., Iosup, A., Prat-Pérez, A., Erling, O., Boncz, P.: Graphalytics: a big data benchmark for graph-processing platforms (2015)
3. Clauset, A., Shalizi, C.R., Newman, M.E.: Power-law distributions in empirical data. SIAM Rev. **51**(4), 661–703 (2009)
4. Erling, O., Averbuch, A., Larriba-Pey, J., Chafi, H., Gubichev, A., Prat, A., Pham, M.-D., Boncz, P.: The LDBC social network benchmark: interactive workload. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, 619–630. ACM (2015)
5. Luc, D.: Non-uniform Random Variate Generation. Springer, New York (1986)
6. Murphy, R.C., Wheeler, K.B., Barrett, B.W., Ang, J.A.: Introducing the graph 500. Cray User's Group (CUG) (2010)
7. Newman, M.E.J.: Power laws, Pareto distributions and Zipf's law. Contemp. Phys. **46**(5), 323–351 (2005)
8. Pérez-Casany, M., Casellas, A.: Marshall-olkin extended Zipf distribution. arXiv preprint (2013). arXiv:1304.4540
9. Yee, T.W.: Maintainer Thomas Yee, and Suggests VGAMdata. Package 'vgam' (2015)