**Control and Cybernetics**

vol. **40** (2011) No. 3

# Random graph generator for bipartite networks modeling*

by

**Szymon Chojnacki and Mieczysław A. Kłopotek**

Institute of Computer Science, Polish Academy of Sciences
Ordona 21, 01-237 Warsaw, Poland

**Abstract:** The purpose of this article is to introduce a new bipartite graph generation algorithm. Bipartite graphs consist of two types of nodes and edges join only nodes of different types. This data structure appears in various applications (e.g. recommender systems or text clustering). Both real-life datasets and formal tools enable us to evaluate only a limited set of properties of the algorithms that are used in such situations. Therefore, artificial datasets are needed to enhance development and testing of the algorithms. Our generator can be used to produce a wide range of synthetic datasets.

**Keywords:** complex networks, random graphs, bipartite graphs, recommender systems, affiliation networks.

## 1. Introduction

The analysis of large networks is driven by the desire to understand and model as diverse phenomena as the spread of infection, social communities creation, protein interactions or website importance assessment (see Newman, 2010). The interest of research community in complex networks was fueled by an empirical evidence which proved that some properties of real-life graphs are unachievable for classic random models. Moreover, similar properties are common to networks observed in various fields. Several statistics describing networks can be measured. However, node degree distribution and mean clustering coefficient are two measures of a great importance. They are related, for example, to such macro features as average length of a path between two nodes, network resilience to an attach or pace of spread of innovations. It turns out that in diverse real-life networks:

- node degree distribution is heavy-tailed
- mean clustering coefficient is bounded away from zero.

*Submitted: March 2011; Accepted: August 2011.

In the classic theory of random graphs developed by Erdős and Rényi (1960) the asymptotic node degree distribution is *Poisson*. Also the value of the clustering coefficient, which measures the probability that two nodes sharing a friend are connected, differs from empirical results.

The seminal paper of Barabási and Albert (1999) describes the driving forces which are responsible for the heavy-tailed node degree distributions. The property can be attributed to both: the growth and the preferential attachment mechanism. Moreover, none of the two results in the desired distribution on its own. Kumar et al. (2000) proposed to substitute the preferential attachment mechanism with random selection of a neighboring node, which also leads to the heavy-tailed distribution. Liu et al. (2002) described how a mixture of preferential and uniform attachment enables us to generate networks with weakened heavy-tail. Vázques (2003) proposed a random graph generation procedure which results in networks with increased values of the clustering coefficient. The combined translation of the four results onto the ground of bigraphs comprises the frame of our algorithm.

Recently, a few random bipartite graph generating algorithms have been introduced: Zheleva et al. (2009), Guillaume and Latapy (2004), Lattanzi and Sivakumar (2009), Chojnacki and Kłopotek (2011b). However, they result in either power-law or exponential node degree distributions. Moreover, the generators do not have parameters responsible for controlling the level of clustering.

Our contribution comprises three main results:

1. definition of a new local clustering coefficient dedicated for bigraphs - the bipartite local clustering coefficient (BLCC)
2. introduction of *bouncing mechanism* responsible for the growth of BLCC
3. description and analysis of a new versatile bigraph generator.

The rest of the article is organized as follows. In Section 2 we formalize node degree distributions, local clustering coefficient and introduce BLCC. In Section 3 we outline potential fields of application for our generator. The fourth section contains a description of our algorithm. In Section 5 we present the results of numerical simulations. The last, sixth section, is dedicated to the concluding remarks. The details of mathematical transformations are given in the appendix.

## 2. Background

A graph is an ordered pair $G = (V, E)$ comprising a set of vertices $V$ and a set of edges $E \subseteq \{V \times V\}$. A bipartite graph has two distinct types of nodes: $U$ and $I$ (i.e. $V = \{U \cup I, U \cap I = \emptyset\}$) and edges exist only between nodes of different types $E \subseteq \{U \times I\}$. We analyze undirected graphs.

### 2.1. Node degree

A degree of a node is the number of direct neighbors of the node. The probability density function of node degree distributions in real-life datasets[1] is usually skewed (see Fig. 2). If the tail decays slowly we can observe the power-law distribution $pdf_{PL}(x) = ax^{-k}$. The tail vanishes quicker in the exponential distribution $pdf_{EX}(x) = \lambda e^{-\lambda x}$. It is convenient to visualize the two distributions on a log-log scale. From the fact that $\log(pdf_{PL}(x)) = -k\log(x) + log(a)$ it follows that the power-law distribution is shaped as a straight line on a log-log chart. This distribution is called *scale-free* because $pdf_{PL}(cx) = a(cx)^{-k} = c^{-k}pdf_{PL}(x)$. The distributions observed in real networks can only rarely be generated by classic random graphs, as it leads to the Poisson distribution. The three types of distributions are drawn in Fig. 1.

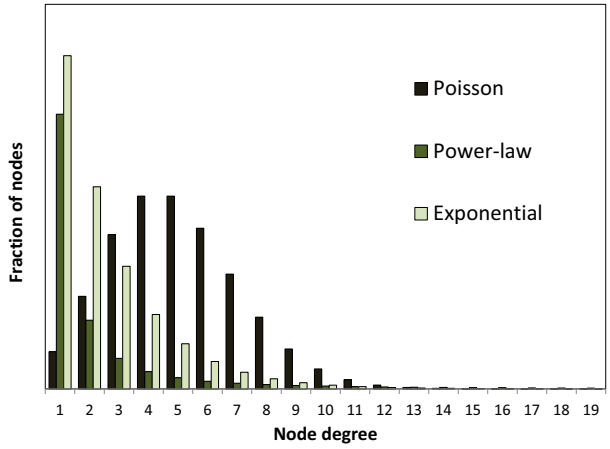


Figure 1. Three degree distributions with the same average. The Poisson distribution is characteristic for classic random graphs. The exponential and the power-law distributions are more common in real datasets. Both of them are skewed. However, the tail of the power-law distribution decays slower

### 2.2. Local clustering coefficient

The local clustering coefficient is used to measure the probability that if two nodes share a neighbor than they are also connected. It is computed for each node and an average value over all nodes indicates the level of the transitivity in a network. Denote by $c_j$ the number of connected pairs among the direct neighbors

[1]The datasets were obtained from three web portals: `http://www.bibsonomy.org`, `http://www.imdb.com` and `http://www.citeulike.org`.

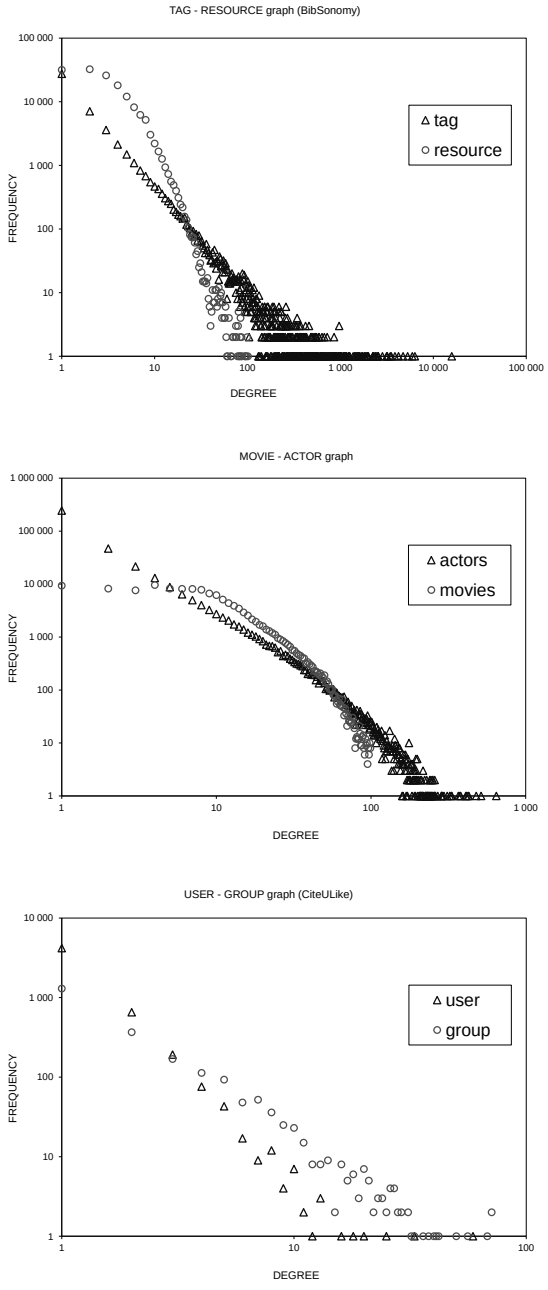


Figure 2. The node degree distributions of three bipartite graphs. The points are plotted with log-log scales

of node $j$ and by $k_j$ the degree of node $j$. The local clustering coefficient ($LCC$) is given by:

$$LCC_j = \frac{c_j}{k_j(k_j-1)/2}. \tag{1}$$

The value of LCC is zero for any node in a bipartite graph. Therefore, we introduce a new coefficient dedicated to measuring transitivity in bigraphs. Bipartite local clustering coefficient ($BLCC$) of node $j$ takes values of one minus the proportion of the node second neighbors to the potential number of the second neighbors of the node. The value of $BLCC$ calculated for node $j$ is given by:

$$BLCC_j = 1 - \frac{|N_2(j)|}{\sum_{i\in N_1(j)}(k_i-1)}, \tag{2}$$

where $|N_2(j)|$ stands for the number of the second neighbors of node $j$, $N_1(j)$ is the set of the first neighbors of node $j$. We show in Fig. 3 that for nodes with a given degree[2] $BLCC$ grows with the number of connected first neighbors.

We also considered a different definition of the number of potential second neighbors in (2). In configuration model (see Newman et al., 2001) it can be approximated with $\langle u\rangle\left(\frac{\langle v^2\rangle}{\langle v\rangle}-1\right)$, where $\langle u\rangle$ and $\langle v\rangle$ are mean degrees of user and item degree distributions, $\langle v^2\rangle$ is the second moment of the item degree distribution. However, BLCC calculated with such approximation can be negative and therefore we stay with the definition of BLCC as in (2). In Table 1 we calculated $BLCC$ for various bipartite graphs. YouTube, Flickr, LiveJournal and Orkut datasets were obtained from Mislove et al. (2007). The CEO dataset was obtained from Wasserman and Faust (1994).

Table 1. $BLCC$ calculated for eight real-life datasets

| | users | items | edges | $BLCC$ |
|---|---|---|---|---|
| CEO | 26 | 15 | 98 | 0.01 |
| CiteULike | 5 208 | 2 336 | 7 196 | 0.41 |
| BibSonomy | 3 617 | 93 756 | 253 366 | 0.92 |
| YouTube | 94 238 | 30 087 | 293 360 | 0.40 |
| IMDB | 383 640 | 127 823 | 1 470 404 | 0.63 |
| Flickr | 395 979 | 103 631 | 8 545 307 | 0.98 |
| LiveJournal | 3 201 203 | 7 489 073 | 112 307 385 | 0.48 |
| Orkut | 2 783 196 | 8 730 857 | 327 037 487 | 0.85 |

[2]In order to visualize the relation we used Enron graph. The dataset was downloaded from `http://snap.stanford.edu/data/`.

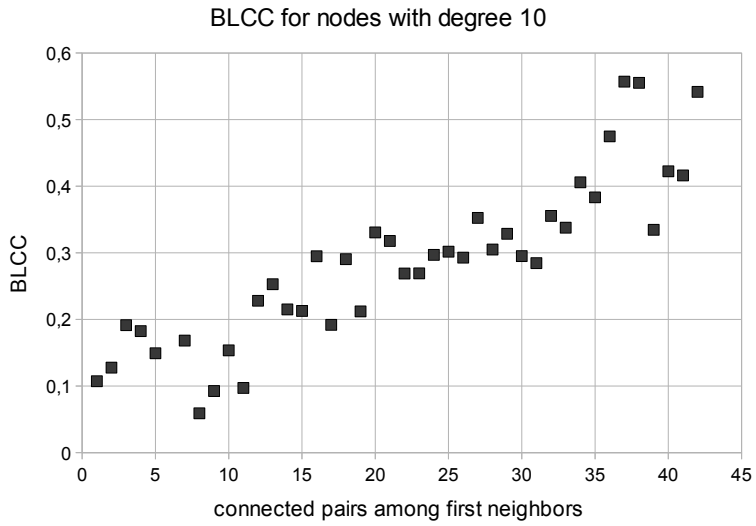


Figure 3. BLCC calculated for nodes with degree 10 in Enron graph

## 3. Potential applications

We describe shortly two scenarios in which our generator can be used. Firstly, by building and measuring random bipartite graphs we can better understand the bipartite structure of a dataset. Secondly, we can produce a wide range of synthetic datasets and compare the performance of various algorithms run on the datasets. Such structure can be observed when clustering text documents. The documents can be perceived as nodes of one type. Words occurring in the documents are nodes of the second type and edges indicate that a word appears in a document. The edges can be weighted with the *tf-idf* formula. The documents are represented as sparse vectors and the task is to find groups of similar vectors. It is very important to understand the structure of the dataset when we have a large corpus and want to do the analysis in a distributed environment. This kind of analysis was recently conducted in the cloud and sponsored by *Amazon*. The details can be found in Potter and Chojnacki (2011).

Bipartite graphs are common in recommender systems. The ratings given to items by users can be stored in a matrix (see Fig. 4). Because such matrix is sparse, it is more convenient to use bipartite graphs to store the dataset (see Fig. 5). It has been shown in Chojnacki and Kłopotek (2011a) that the generator described in this paper can be used to evaluate technical properties of recommender systems. It was used to measure latency, memory consumption, time needed to refresh a model and time of training a model.

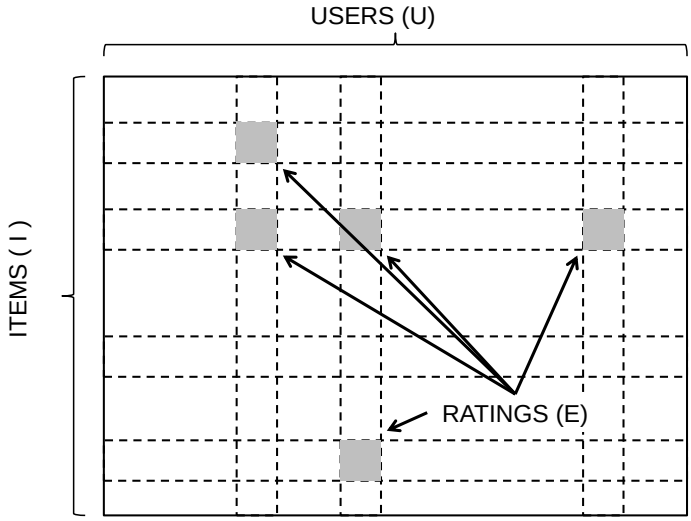


Figure 4. User-Item matrix with historic ratings

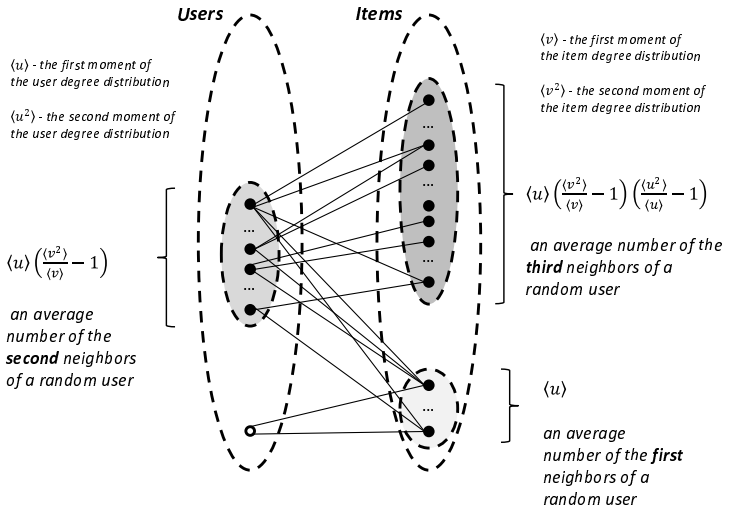


Figure 5. Bipartite graph representation of a user-item matrix

## 4. Our algorithm

Our algorithm consists of three steps: (1) new node creation, (2) edge attachment type selection and (3) running the bouncing mechanism. We need to set eight parameters in the procedure:

- $m$ – the number of initial loose edges with a user and an item at the ends
- $T$ – the number of iterations
- $p$ – the probability that a new node is a user, $(1-p)$ is the probability that a new node is an item
- $u$ – the number of edges created by each new user
- $v$ – the number of edges created by each new item
- $\alpha$ – the probability that a new user edge is being connected to an item with preferential attachment
- $\beta$ – the probability that a new item edge is being connected to a user with preferential attachment
- $b$ – the fraction of preferentially attached edges that were created via a *bouncing mechanism*

Steps (1) and (2) are explained in Section 4.1 and analyzed in Section 4.2. In Section 4.3 step (3) is discussed.

### 4.1. Basic model

In the basic model we utilize first seven parameters. The bouncing mechanism is applied in the full model as an additional third step. The basic model is based on an iterative repetition of two steps.

**Step 1** If a random number is greater than $p$ create a new user with $u$ loose edges, otherwise create a new item with $v$ loose edges.

**Step 2** For each edge decide whether to join it to a node of the second modality randomly (uniformly) or with preferential attachment. The probability of selecting preferential attachment is $\alpha$ for a new user and $\beta$ for a new item.

### 4.2. Formal analysis

At the beginning we have $m$ users. If we run $t$ iterations, the number of users grows to $|U(t)| = m + pt$. After $t$ iterations we have $|I(t)| = m + (1-p)t$ items and $|E(t)| = m + t(pu + (1-p)v)$ edges. The number of edges created in one step is on average $\eta = (pu + (1-p)v)$. When we neglect $m$, the average user degree does not depend on time:

$$\frac{|E(t)|}{|U(t)|} = \frac{m + t(pu + (1-p)v)}{m + pt} \approx \frac{\eta}{p},$$

the average item degrees can be approximated with:

$$\frac{|E(t)|}{|I(t)|} \approx \frac{\eta}{(1-p)}.$$

In the following, we look from user modality perspective. However, the computations can be altered to the opposite item modality easily. In order to derive asymptotic node degree distribution in our model we need to specify the probability that a user node $j$ with degree $k_j$ gets connected to a new item. The quantity is usually represented as $\Pi(k_j)$ within the complex networks community. If nodes are selected uniformly, then:

$$\Pi_{random}(k_j) = \frac{1}{|U(t)|} = \frac{1}{pt}.$$

In case of the uniform attachment $\Pi(k_j)$ does not depend on $k_j$. If nodes are selected with accordance to the preferential attachment rule than:

$$\Pi_{preferential}(k_j) = \frac{k_j}{|E(t)|} = \frac{k_j}{\eta t}.$$

Contrary to the uniform attachment scenario, the probability of node selection is linearly proportional to its current degree. The probability of drawing a node with degree $k_j$ is the degree divided by the number of edges. We can verify that by summing the values of $\Pi$ over all user nodes we get one. In our model the decision whether to draw a user for an item with uniform or preferential attachment depends on $\beta$, hence the combined formula is:

$$\Pi(k_i) = \beta\frac{1}{pt} + (1-\beta)\frac{k_i}{\eta t}. \tag{3}$$

Equation (3) allows for describing the pace of growth of nodes with degree $k_i$ as: vspace-1ex

$$\frac{\partial k_i}{\partial t} = (1-p)v\Pi(k_i). \tag{4}$$

We assume that time intervals between iterations are equal and very small. Moreover, we assume that all nodes with a given degree grow at the same pace and are continuous. We show in the appendix that:

$$P(k) \propto \left(\frac{\beta\eta + p(1-\beta)k}{\beta\eta + p(1-\beta)u}\right)^{\frac{-\eta}{(1-p)(1-\beta)v}-1}. \tag{5}$$

For $\beta = 0$ we get the power-law distribution. If $\beta \to 1$, we can utilize the fact that $\lim_{n\to\infty}\left(1+\frac{c}{n}\right)^n = e^c$ to obtain the exponential distribution. When we put $\beta = 0$, $p = 0.5$ and $u = v$ we have power-law distribution with the scaling exponent equal to 3. The above result is consistent with Barabási and Albert (1999).

### 4.3. Full model

In order to parameterize the transitivity of a network we introduce the *bouncing mechanism* (Fig. 6). It is based on surfing the web technique by Vázquez (2003).

The mechanism enables us to increase the BLCC, but can only be applied to the edges that are to be selected with preferential attachment. This can be attributed to the fact that the probability that a random walk terminates in a node is proportional to its degree, Burda et al. (2009). Bouncing consists of three small steps: (1) a random node is drawn from the nodes that are already joined with the new node, (2) a random neighbor of the drawn node is chosen, (3) a random neighbor of the neighbor is selected for joining with the new node. The pseudo-code of full model is given in Algorithm 1.

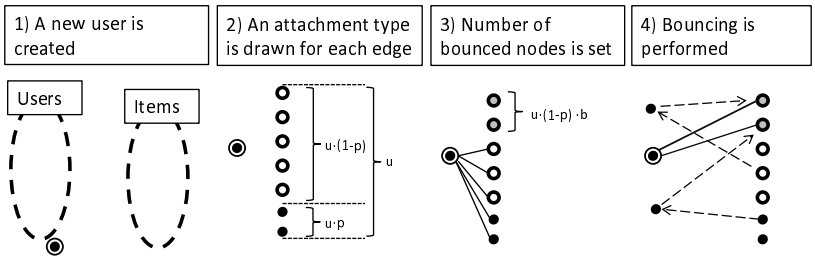


Figure 6. For each edge of a new node that is to be connected with an existing node in accordance to the preferential attachment mechanism, a decision is made whether to create it via a bouncing mechanism. In case of attaching new user node, $u$ new edges are created. On average $u \cdot \alpha$ edge endings are to be drawn preferentially and $u \cdot \alpha \cdot b$ of them are to be obtained via bouncing from the nodes that are already selected

## 5. Numerical results

The results of the numerical experiments are divided into three subsections. In the first part we shortly present a Java applet developed in our Lab to play with various parameters of the generator. In the second part we show which parameters impinge on the shapes of node degree distributions and the values of BLCC. In the last section we show how the number of potentially similar users and the number of their items can be determined by various levels of the generator parameters.

### 5.1. Graphical analysis

The applet presented in Fig. 7 can be accessed online in `http://www.ipipan.eu/~sch/software/applet.html`. All parameters (except for the initial number of pairs) can be changed during graph generation. The distributions of BLCC and node degrees are being updated online for both modalities. Also the average number of potentially similar users and their items is visualized with a chart. By an expression *similar user* we understand all users that have rated at least one item in common with the selected user.

**Algorithm 1:** An iteration of the bipartite graph generator

```
if RAND() ≤ p then
    // p - the probability that a new node is a user
    for k ← 1 to u do
        // u - the number of edges created by anew user
        if RAND() ≤ α then
            // α - the probability that the new user's item is
               drawn preferentially
            if RAND() ≤ b then
                // b - the probability that new preferential node
                   was chosen by bouncing
                SelectedItem ← BounceFromRandom(TempItems) ;
            else
                SelectedItem ← DrawItemPreferentially() ;
                TempItems ← SelectedItem ;
        else
            SelectedItem ← DrawItemRandomly ;
            TempItems ← SelectedItem;
    Users ← Users ∪ NewUser;
    Edges ← Edges ∪{TempItems × NewUser} ;
else
    Process analogously with new item node
```

### 5.2. Social network properties

We consider node degree distributions of both modalities and the values of BLCC as the social network properties of the generated graphs. Node degree distributions are controlled by two parameters: $\alpha$ and $\beta$. We show in Fig. 8 that if any of the two parameters tends to one, the shape of an appropriate modality becomes power-law. Low values of parameters yield exponential distribution.

The values of BLCC can be controlled by the extent of the bouncing mechanism (Fig. 9). If we neglect the bouncing mechanism ($b = 0$), then BLCC is controlled by the node degree distributions (Fig. 10).

There exist several other network properties that can be tuned by the parameters in our model, such as the average distance between randomly selected pairs of nodes, the diameter of a bigraph, resilience to an attack, spread of innovations or creation of the largest connected component. We omit the analysis of these features as they do not seem to have direct impact on the performance of the recommender systems.

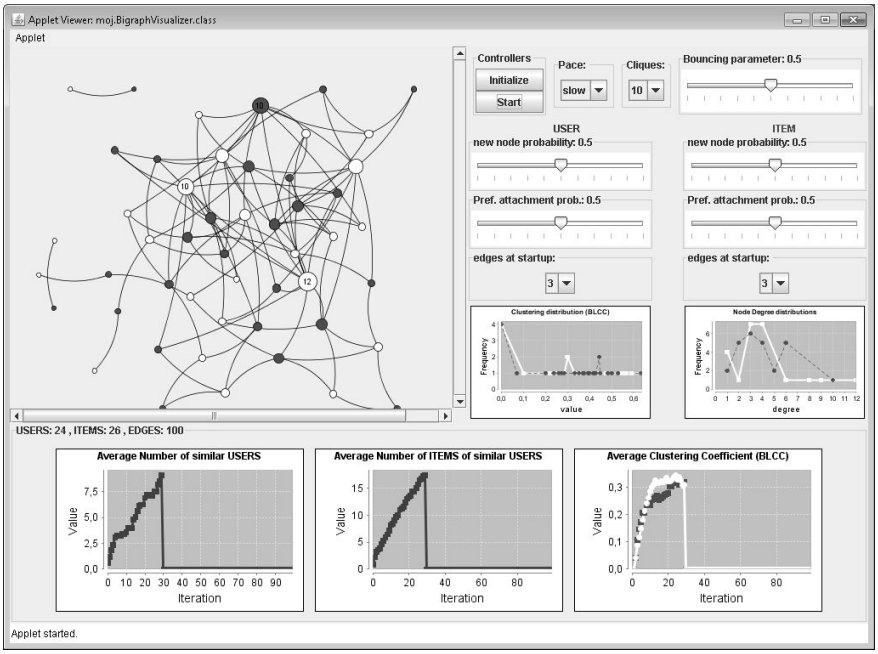


Figure 7. A bigraph generated after $t = 30$ iterations. The values of all probabilities were set to 0.5, each new node creates three new edges $u = v = 3$, initial number of pairs $m = 10$

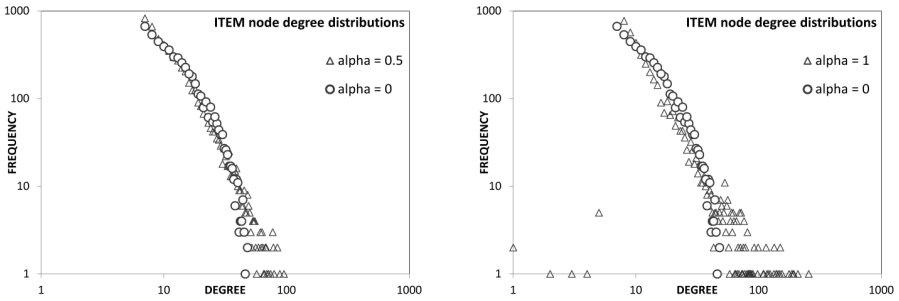


Figure 8. Left panel: circles indicate that the random attachment of users' edges (i.e. items) results in the exponential distribution of item degrees. Triangles in both panels show that as $\alpha \to 1$ the distribution becomes power-law. Experiments run with $m = 50$, $T = 10\,000$, $p = 0.5$, $u = v = 7$, $\beta = 0.5$

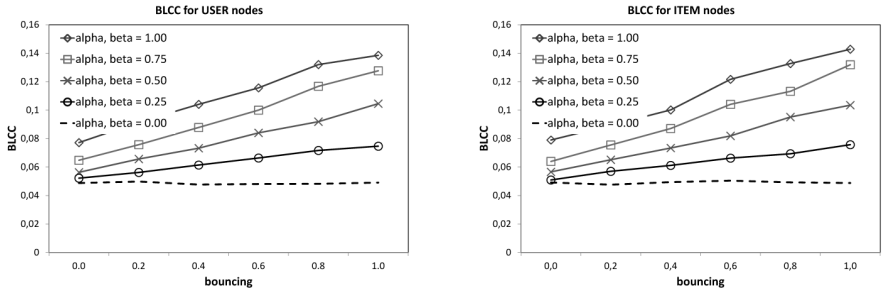


Figure 9. The increase of the bouncing parameter $b$ results in higher values of BLCC (bipartite local clustering coefficient). If no nodes are connected in accordance to the preferential attachment mechanism $\alpha = \beta = 0$, the values of $b$ do not influence BLCC. Experiments run with $m = 50$, $T = 10\ 000$, $p = 0.5$, $u = v = 7$

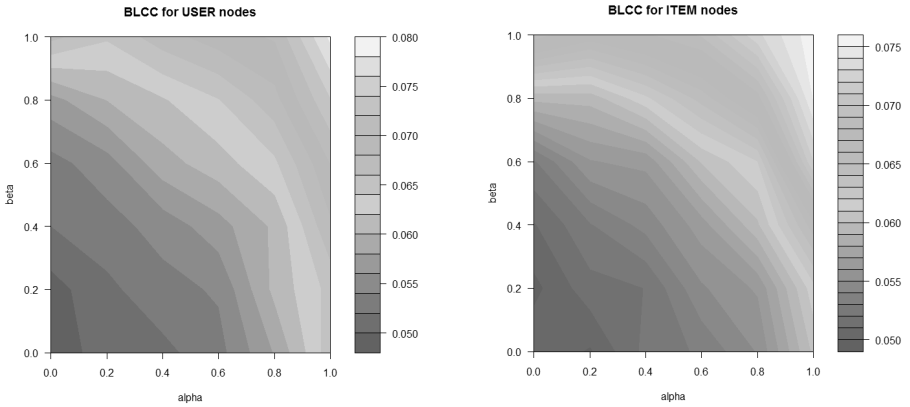


Figure 10. BLCC increases as more edges are connected with preferential attachment mechanism. The phenomenon is observed even when the bouncing parameter is zero. Experiments run with $m = 50$, $T = 10\ 000$, $p = 0.5$, $u = v = 7$, $b = 0$

### 5.3. Size of the neighborhood

The number of operations that a neighborhood recommender system has to perform is related to the number of similar users and the number of their items. We recommend a new item to an analyzed user from the items of the users that are similar to her/him. In Fig. 11 we show two results:

- the size of the neighborhood grows with the size of a graph
- the size of the neighborhood grows with the density of a graph (fixed number of nodes and growing number of edges).

The growth of the neighborhood is relatively sharper in case of the number of items. It is interesting that the number of similar users becomes stable earlier for sparser graphs (3 and 6 edges at startup) than for denser graphs (12 and 24 edges at startup).

A result of potentially great importance is drawn in Fig. 12. It turns out that the impact of the shapes of node degree distributions (controlled by parameters $\alpha$ and $\beta$) on the sizes of the neighborhoods is not monotonic. The more exponential - like than power-law-like the distribution of users' degrees the smaller number of similar users is observed. In all other cases the opposite influence is observed.

The result presented in Fig. 13 is somewhat disappointing. The shrinking impact of the bouncing mechanism on the sizes of the neighborhoods is hardly observed. The effect of bouncing is too weak compared to the level at which we are placed by the power-law distribution. Also random changes among various networks are stronger at the level than the shrinking forces. This drawback reflects the fact that in growing random graphs positive clustering coefficient is correlated with power-law node degree distribution and we are unable to generate graphs with both the exponential node degree distribution and high value of clustering.

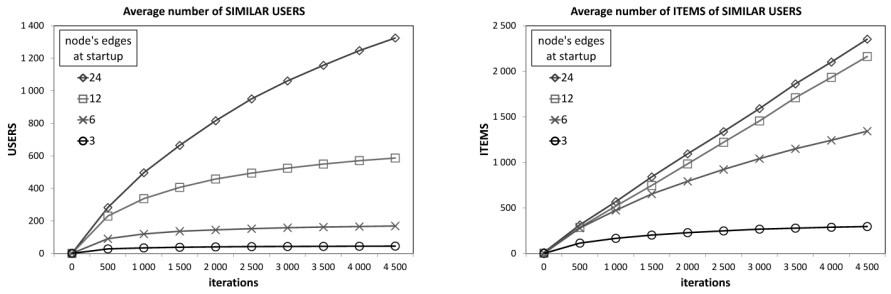


Figure 11. The average number of similar users (having at least one common item with a considered user) follows the increase in graph size. The positive relation is stronger in case of the number of items of similar users. The density of a graph (modeled by the number of startup edges) has even stronger impact on the size of the neighborhood than the size of a graph. Experiments run with $m = 50$, $T = 10\ 000$, $p = 0.5$, $\alpha = \beta = 0.5$, $b = 0$

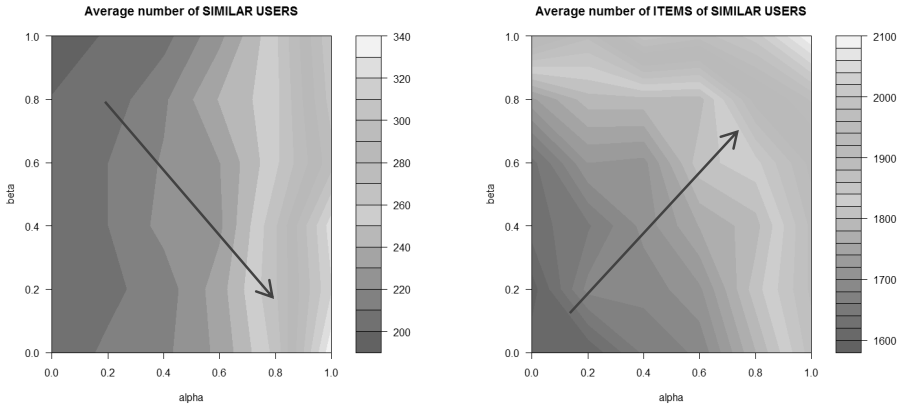


Figure 12. The shape of node degree distributions of both modalities has opposite influence on the average number of similar users. The more power-law-like item degree distribution, the more neighbors can be observed. The more heavy-tailed the distribution of user nodes the stronger shrinking of the neighborhood is obtained. The arrows indicate the direction of increase. Experiments run with $m = 50$, $T = 10\ 000$, $p = 0.5$, $u = v = 7$, $b = 0$

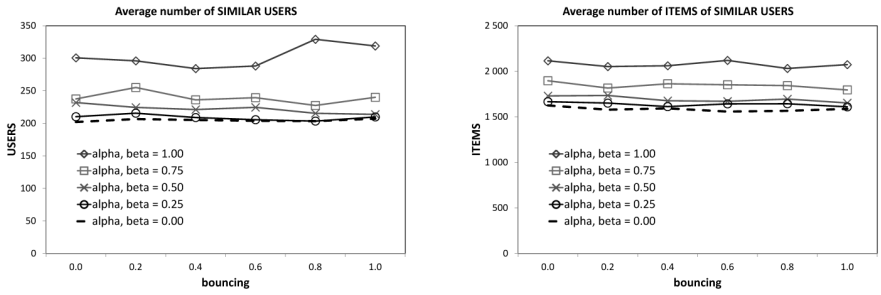


Figure 13. Increase of the bouncing parameter $b$ has a slight negative impact on the size of both neighborhoods. However, the number of similar users and their items is determined mostly by the shapes of node degree distributions

## 6. Conclusion

We have presented a new random graph generation algorithm dedicated to modeling performance of recommender systems. We have shown that the parameters of the algorithms influence not only pure network properties of created bigraphs, but also the properties related to the performance of neighborhood based collaborative filtering systems. Besides the above features, the procedure enables

us to output bigraphs of different sizes, densities and proportions of the number of users to the number of items. We plan to compare how various features of bigraphs impinge on time and memory requirements of existing systems. Consequently, to better understand the algorithms, their implementations and finally to improve both of them.

**Acknowledgments**

This work was partially supported by Polish state budget funds for scientific research within the research project *Analysis and visualization of structure and dynamics of social networks using nature inspired methods*, grant No. N516 443038.

## References

Barabási, A.L. and Albert, R. (1999) Emergence of Scaling in Random Networks. *Science*, **286**(5439), 509–512.

Burda, Z., Duda, J., Luck, J.M. and Waclaw, B. (2009) Localization of the Maximal Entropy Random Walk. *Phys. Rev. Lett.*, **102**(16), http://prl.aps.org/abstract/PRL/v102/i16/e160602.

Chojnacki, Sz. and Kłopotek, M.A. (2011a) Random Graphs for Performance Evaluation of Recommender Systems. *Control and Cybernetics*, **40** (2), 237–257.

Chojnacki, Sz. and Kłopotek, M.A. (2011b) Scale invariant bipartite graph generative model. In: *International Joint Conference Security and Intelligent Information Systems*. **LNCS 7053**, Springer.

Erdős, P. and Rényi, A. (1960) On the Evolution of Random Graphs. In: *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, 17–61.

Guillaume, J.-L. and Latapy, M. (2004) Bipartite structure of all complex networks. *Inf. Process. Lett.*, **90**(5), 215–221.

Kumar, R., Raghavan, P., Rajagopalan, S., Sivakumar, D., Tomkins, A. and Upfal, E. (2000) Stochastic Models for the Web Graph. In: *Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE CS Press, Redondo Beach, CA, USA, 57–65.

Lattanzi, S. and Sivakumar, D. (2009) Affiliation networks. In: *STOC '09: Proceedings of the 41st annual ACM symposium on Theory of computing*. ACM, New York, NY, USA, 427–434.

Liu, Z., Lai, Y.Ch., Ye, N. and Dasgupta, P. (2002) Connectivity distribution and attack tolerance of general networks with both preferential and random attachments. *Physics Letters A*, **303**(5-6), 337–344.

Mislove, A., Marcon, M., Gummadi, K.P., Druschel, P. and Bhattacharjee, B. (2007) Measurement and Analysis of Online Social Networks. In: *Proceedings of the 5th ACM/USENIX Internet Measurement Conference (IMC'07)*, San Diego, CA.

Newman, M. (2010) *Networks: An Introduction*, Oxford University Press.
Newman, M., Strogatz, S. and Watts, D.J. (2001) Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, **64**, 2.
Potter, T. and Chojnacki, Sz. (2011) Mahout Clustering Benchmarks. In: G.S. Ingersoll, T.S. Morton and A.L. Farris, eds., *Taming Text. How to Find, Organize, and Manipulate It*. Manning, 211–216.
Vázquez, A. (2003) Growing network with local rules: Preferential attachment, clustering hierarchy, and degree correlations. *Phys. Rev. E*, **67**(5).
Wasserman, S. and Faust, K. (1994) *Social Network Analysis: Methods and Applications*. Cambridge University Press.
Zheleva, E., Sharara, H. and Getoor, L. (2009) Co-evolution of social and affiliation networks. In: J.F. Elder IV, F. Fogelman-Soulié, P.A. Flach and M. Zaki, eds., *KDD*. ACM, 1007–1016.

## A. Node degree distribution

We follow *continuum approach* of Barabási and Albert (1999) to derive user node degree distribution. The item node degree distribution can be obtained analogously. The calculations consist of three steps. Firstly, let us solve Eq. (4):

$$\begin{aligned}\frac{\partial k_j}{\partial t} &= (1-p)v\Pi(k_j)\\ &= (1-p)v\left(\frac{\beta}{pt}+\frac{(1-\beta)k_j}{\eta t}\right)\\ &= (1-p)v\frac{1}{t}\left(\frac{\beta\eta+p(1-\beta)k_j}{p\eta}\right).\end{aligned}$$

It yields:

$$\int \frac{1}{(1-p)v}\cdot\frac{p\eta}{\beta\eta+p(1-\beta)k_j}\,dk_j=\int\frac{1}{t}\,dt. \tag{6}$$

Taking into account an initial condition $k_j(t_j)=u$, where $t_j$ is the time of creating user $j$, and the fact that $\int\frac{c}{ax+b}dx=\frac{c}{a}\ln|ax+b|+C$, we obtain:

$$\frac{p\eta}{(1-p)vp(1-\beta)}\left(\left[\ln\left(\beta\eta+p(1-\beta)k_j\right)\right]-\left[\ln\left(\beta\eta+p(1-\beta)u\right)\right]\right)=\left[\ln t\right]-\left[\ln t_j\right], \tag{7}$$

both sides of which can be used as exponents of $e$, giving:

$$\left(\frac{\beta\eta+p(1-\beta)k_j}{\beta\eta+p(1-\beta)u}\right)^{\frac{\eta}{(1-p)(1-\beta)v}}=\left(\frac{t}{t_j}\right). \tag{8}$$

After reorganizing, we have:

$$k_j(t) = \frac{1}{p(1-\beta)} \cdot \left( (\beta\eta + p(1-\beta)u) \left( \frac{t}{t_j} \right)^{\frac{(1-p)(1-\beta)i}{\eta}} - \beta\eta \right). \tag{9}$$

The probability that $k_j$ is smaller than a given $k$ is:

$$\Phi\left\{k_j(t) < k\right\} = \Phi\left\{ \frac{(\beta\eta + p(1-\beta)u) \left( \frac{t}{t_j} \right)^{\frac{(1-p)(1-\beta)v}{\eta}} - \beta\eta}{p(1-\beta)} < k \right\}, \tag{10}$$

and after reorganizing:

$$\Phi\left\{k_j(t) < k\right\} = \Phi\left\{ t_j > t \left( \frac{\beta\eta + p(1-\beta)k}{\beta\eta + p(1-\beta)u} \right)^{\frac{-\eta}{(1-p)(1-\beta)v}} \right\}. \tag{11}$$

We have assumed that nodes are added at equal time intervals until the current iteration $t$. The probability that the iteration of adding node $j$ is larger than some $K \leq t$ equals $1 - \Phi(t_j \leq K) = 1 - K\frac{1}{t}$. Substituting this assumption into Eq. (11), we obtain:

$$\begin{aligned} \Phi\left\{k_j(t) < k\right\} &= 1 - \Phi\left\{ t_j \leq t \left( \frac{\beta\eta + p(1-\beta)k}{\beta\eta + p(1-\beta)u} \right)^{\frac{-\eta}{(1-p)(1-\beta)v}} \right\} \\ &= 1 - \left( \frac{\beta\eta + p(1-\beta)k}{\beta\eta + p(1-\beta)u} \right)^{\frac{-\eta}{(1-p)(1-\beta)v}}. \end{aligned}$$

We can obtain probability density function of random variable $k$ by differentiating its cumulative distribution function $P(k) = \partial\Phi\{k_j(t) < k\}/\partial k$, as a result we have:

$$P(k) = \frac{\eta}{(1-p)(1-\beta)v} \cdot p(1-\beta) \cdot \left( \frac{\beta\eta + p(1-\beta)k}{\beta\eta + p(1-\beta)u} \right)^{\frac{-\eta}{(1-p)(1-\beta)v} - 1}, \tag{12}$$

that is:

$$P(k) \propto \left( \frac{\beta\eta + p(1-\beta)k}{\beta\eta + p(1-\beta)u} \right)^{\frac{-\eta}{(1-p)(1-\beta)v} - 1}. \tag{13}$$