Exploiting Phase Transitions for the Efficient Sampling of the Fixed Degree Sequence Model

Christian Brugger, André Lucas Chinazzo,
Alexandre Flores John,
Christian De Schryver, Norbert Wehn
Microelectronic System Design Research
Group, University of Kaiserslautern, Germany
Email: {brugger, schryver, wehn}@eit.uni-kl.de,
{chinazzo, floresj}@rhrk.uni-kl.de

Andreas Spitz
Database Systems Research
Group, Heidelberg
University, Germany
Email: spitz@informatik.
uni-heidelberg.de

Katharina Anna Zweig
Complex Network Analysis
Group, University of
Kaiserslautern, Germany
Email: zweig@cs.uni-kl.de

***Abstract*—Real-world network data is often very noisy and contains erroneous or missing edges. These superfluous and missing edges can be identified statistically by assessing the number of common neighbors of the two incident nodes. To evaluate whether this number of common neighbors, the so called *co-occurrence*, is statistically significant, a comparison with the expected co-occurrence in a suitable random graph model is required. For networks with a skewed degree distribution, including most real-world networks, it is known that the *fixed degree sequence model*, which maintains the degrees of nodes, is favourable over using simplified graph models that are based on an independence assumption. However, the use of a fixed degree sequence model requires sampling from the space of all graphs with the given degree sequence and measuring the co-occurrence of each pair of nodes in each of the samples, since there is no known closed formula for this statistic. While there exist log-linear approaches such as Markov chain Monte Carlo sampling, the computational complexity still depends on the length of the Markov chain and the number of samples, which is significant in large-scale networks. In this article, we show based on ground truth data that there are various phase transition-like tipping points that enable us to choose a comparatively low number of samples and to reduce the length of the Markov chains without reducing the quality of the significance test. As a result, the computational effort can be reduced by an order of magnitudes.**

I. Introduction

The identification of so-called *network motifs*, i.e., subgraphs whose occurrence is statistically significant, is of general interest, especially in biological data sets [1]. The significance of a subgraph is tested by counting its occurrence in an observed real-world network and by comparing it to the expected occurrence in a suitable random graph model. A *random graph model* is defined as a set of graphs together with a probability mass function that assigns a probability to each member of the set, summing up to 1. While not specifically a network motif, the number of common neighbors of two nodes x, y, their so-called *co-occurrence* $coocc(x, y)$, can also be tested on its statistical significance in the same way. Zweig and Kaufmann proved that a simple independence model, which estimates the expected co-occurrence of two nodes to be $deg(x)deg(y)/2m$, where $deg(x)$ denotes the *degree* of node x, is wrong if the degree sequence is skewed [2], [3]. The *degree sequence* $DS(G)$ of a graph G is defined as the sequence of degrees of the nodes of G in some fixed order. Thus, a more detailed random graph model has to be used, such as the set $\mathcal{G}(DS)$ of all simple graphs with the same degree sequence as the observed network and uniform probability. In the following, this model is called the *fixed degree-sequence model* or **FDSM**. Note that a graph is *simple* if it does not contain multiple edges between the same nodes and no self-loops.

The statistical significance of the number of common neighbors can be used to do *link assessment*, i.e., to evaluate whether an existing edge in a graph is likely to be a true-positive and whether a non-existing edge is likely to be a false-negative [4], [5], [6]. A link assessment results in a ranking of all pairs of nodes, where (existing) edges with a high ranking are assumed to be true-positives and pairs of nodes that are ranked highly but are not yet connected by an edge are considered to be false-negatives. This assumption can be quantified for networks with an assigned *ground truth*, i.e., networks for which there is a noisy edge set containing false-positives and false-negatives, and a verified set of edges.

Unfortunately, $\mathcal{G}(DS)$ is far to large to enumerate it. Sampling from $\mathcal{G}(DS)$ therefore is the only viable option of obtaining an estimate of the expected co-occurrence of two nodes. Del Genio et al proposed an *exact sampling scheme*, which constructs graphs with an arbitrary degree sequence with uniform probability [7]. However, the complexity of this algorithm is in $\mathcal{O}(n^3)$, making it an infeasible choice for the repeated sampling of large-scale networks. As a result, a Markov chain Monte Carlo approach is frequently used in practice, even though the exact *mixing time* and therefore the computational complexity are unknown. In this article, we only consider undirected, bipartite networks and thus give an overview of the Markov chain for such graphs in the following [8]. Analogous approaches exist for non-bipartite and even directed graphs [9]. Starting from the observed network, in every step two edges $e_1 = (a, b)$ and $e_2 = (c, d)$ are chosen uniformly at random from all edges and it is tested whether they are swappable, i.e., whether $e'_1 = (a, d)$ and $e'_2 = (c, b)$ are already in the graph. If they are swappable, the tested edges e'_1 and e'_2 are inserted into the graph and e_1, e_2 are deleted. A pre-defined number of swap tests are done—regardless of the result of the test—and it can be shown that after a sufficient number of tests, the resulting graph is any graph from $\mathcal{G}(DS)$ with uniform probability. The sufficient number, the so-called *mixing time*, is to date unknown for this specific Markov chain—and the known upper bounds (e.g., [10], [11]) are of little practical relevance. They are either too big like Jerrum et al's result in $O(n^{14}\log^4 n)$ or not computable for large networks like Brualdi's result, who showed that the convergence time depends on the spectral gap of the transition matrix of the Markov Chain [12]. Computing the latter would require to know all possible graphs in $\mathcal{G}(DS)$ and their transitions.

Depending on the size of the graph and the required data structures to store it in memory, a single swap test and updating the data structure(s) after a successful swap cost between $O(1)$ and $O(\log n)$ or $O(\min\{deg(x), deg(y)\})$. Note that, as for all Markov chains with known degrees, there is the probability of an importance sampling, but again, it is infeasible for large data sets [13].

A *safe* number of steps is considered to be in $O(m \log m)$. It is the lower bound such that, expectedly, every edge is chosen at least once for a swap test. Often, the safe number of steps is also used for a so-called *burn-in* phase where one tries to move away from the often very strongly structured observed network to one that is more random. From this instance, a chain of swaps is started, where every x-th resulting graph is tested for its structural features—these graphs comprise the *sample* against which the statistical significance of structural features of the observed graph is tested. Again, a "safe" size of this set is often used, for example, $10,000$ samples. This scheme will be called the *serial burn-in (sampling) scheme* in the following. To empirically analyze the necessary burn-in length, Gionis et al. proposed empirical convergence tests based on practices from data mining, namely observing the convergence of the number of so-called *frequent item sets* or their frequencies [14], [13]. By plotting the number of frequent item sets in dependence of the number of swap tests, it can be seen that this number stabilizes (Gionis et al., Fig. 4). However, counting frequent item sets is computationally expensive and Gionis et al. do not provide an online stopping criterion that allows to stop sampling.

This article looks at link assessment in bipartite graphs, i.e., given a graph $G = (V_L \cup V_R, E)$ with a node set V_L, a node set V_R and an edge set E connecting nodes of V_L with nodes of V_R, we assess whether any two nodes in V_R have a statistically significant number of neighbors in common. This information can then be used to build insightful one-mode projections of bipartite graphs [2], [3], [15]. In this article we provide two online heuristics, one to determine a sufficient number of the number of swaps, and one to determine a sufficient number of samples. For the first time, the quality of the resulting statistics is tested against ground truth for the link assessment task which shows astonishing *tipping points*: while at first—with low numbers of swap tests—the link assessment is not good, a small increase has a strong impact on the quality of the link assessment—an effect that is often called a *phase transition*. Similarly, computing the co-occurrences for all pairs of nodes of interest in one sample is costly (in $\Omega(n^2)$ to $O(n^3)$, depending on data structures and density). Thus, reducing the number of samples is also of interest and, again, we find that there is a phase-transition-like behavior in the number of samples. Finally, especially for bipartite graphs in which the hidden connections between nodes on one side of the graph are assessed [2], [3], [15], it can be beneficial to reduce the graph by sampling from nodes of the other side. For example, in market-basket data with millions of customers but only thousands of products, it might not be necessary to look at the whole data set but to reduce it to the market baskets of $50,000$ customers. Again, we find a phase-transition like behavior in this case as well that points to an optimal set. By optimizing these parameters, we achieve speedups of up to one order of magnitude.

Our novel contributions are:

- We show that there is a phase transition-like behavior in the number of swaps, the number of samples, and the sample size of the left-hand side in a bipartite graph.
- We present two online heuristics for estimating just the required number of swaps and number of samples.
- We demonstrate the effectiveness and stability of our heuristics in empirical studies for multiple data sets.

The next section introduces definitions from statistics to assess the significance of the co-occurrence of two nodes in a bipartite graph.

II. Definitions

Given a bipartite graph $G = (V_L \cup V_R, E)$ as defined above, the co-occurrence of two nodes $x, y \in V_R$ is defined as the number of their common neighbors in V_L. This value is bounded from above by $\min\{deg(x), deg(y)\}$, the minimal degree of both nodes. Thus, its absolute value cannot be used to understand its significance, since nodes with a small degree would always be disfavored. Their *expected* co-occurrence with respect to some random graph model, e.g., the $FDSM$ ($coocc_{FDSM}(x, y)$), is defined as the expected co-occurrence in all graphs in the model, given their probability. If it is not possible to compute it, it is approximated by the average observed co-occurrence in a uniform sample from the random graph model. Note that the approximation quality depends on the sample size and on the quality of the sample, as introduced in [2]. For a sample from a random graph model, two statistical measures can be computed with respect to a given pair of nodes and their observed co-occurrence. The p-value denotes the fraction of observed samples in which the co-occurrence of x and y was at least as high as the observed one; the z-score of the observed co-occurrence is given by the co-occurrence distribution of x and y in the sample:

$$p\text{-}value(x, y) = \sum_{i=1}^{s} \begin{cases} 1, & \text{if } coocc_i(x, y) > coocc(x, y) \\ 0, & \text{otherwise} \end{cases},$$

$$z\text{-}score(x, y) = \frac{leverage(x, y)}{\text{stddev}\left(\{coocc_i(x, y)\}_{i=1,\ldots,s}\right)},$$

where s is the number of samples and leverage is defined as in [2]. Based on previous work by Zweig et al. [2], [3] and Horvát et al. [15], two nodes with a high z-score, or low p-value, are connected in a one-mode projection of the bipartite graph. It often shows that they are also semantically similar and this method has been used to identify similar movies [2], [3] or biologically similar proteins [4]. In this article, we rank all edges by their p-value and break ties by the z-score.

III. Ground truth and PPV_k

We are using two data sets, the Netflix competition data set and a medium size MovieLens data set[1]; both data sets show ratings of films by a number of users. By setting a threshold, the data can be represented as a bipartite graph between users and movies, where an edge (u, j) represents that user u likes film j. By finding significant co-occurrences between any two movies i, j, a one-mode projection can be obtained [3]. We use a ground truth data set based on movie sequels such as Star Wars or James Bond, first used in a paper by Horvát and Zweig [6] which can be used for both data sets. For a given set of movie sequels, the idea is that the most significant co-occurrences are assumed to be with other sequels from the same set. Thus, ranking all pairs of films (where at least one is a sequel from a series) by the significance of their co-occcurrence, the most significant pairs should be sequels from the same set. The quality of such a ranking can be evaluated by the *positive predictive value at k* PPV_k, where the k indicates the number of pairs of films in the ground truth and the PPV is the fraction of correctly identified pairs from the ground truth in the set of the k highest ranked pairs of films. This measure was proposed by Liben-Nowell and Kleinberg as more meaningful in the very unbalanced link prediction problem, which is very similar to the problem proposed here [16].

IV. Phase Transitions

For each parameter of the algorithm, i.e., the number of swaps, the number of samples and the length of the burn-in-phase, the PPV_k can be computed for the resulting ranking of the co-occurrences. Here we show that the quality of the ranking improves very suddenly in all of the three parameters. This indicates on the one hand that these parameters are often smaller than anticipated, but also that they have to be chosen carefully since the quality does not rise linearly in any of them: choosing too low a number of, e.g., swaps can significantly harm the quality of the algorithm. A final question to be analyzed is whether it is reasonable to take the full Netflix data set with 100 million ratings or if a random subset of the data is good enough.

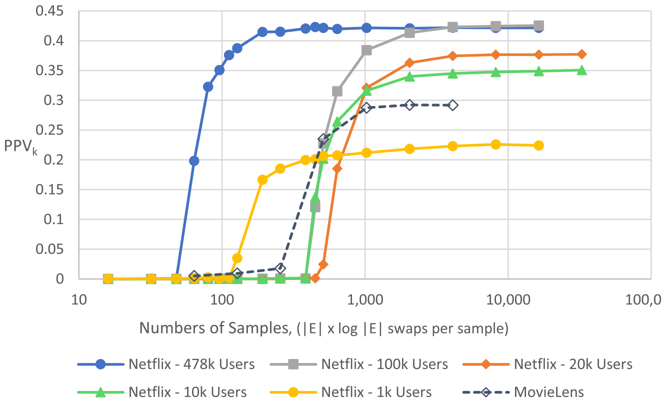

Fig. 1. Quality over number of samples. For a wide variety of data sets, narrow phase transitions are present.

For the runtime of the algorithm, the most important factor is the number of samples, since the co-occurrence has to be computed for every pair of movies of interest which - in general - is in $O(n^3)$. Fig. 1 shows a very steep transition in quality in dependence of the number of samples: The full MovieLens data set requires about 1024 samples to reach a PPV_k-value of 0.286 ± 0.005. Spending more samples only marginally improves the results, e.g., for 4096 samples the PPV_k-value is 0.291 ± 0.002. For the full Netflix data set, 384 samples already result in a PPV_k-value of 0.4206 ± 0.0019, while $16,384$ samples only improve this value to 0.4217 ± 0.0012, but would take 43 times longer to compute. Note, however, that this steep, phase-transition-like behavior also has the downside that too small a number of samples decreases the quality enormously: using 64 instead of 384 samples still yields a PPV_k of 0.20 ± 0.03. However, using 48 samples decreases it to a meaningless value of 0.001 ± 0.001.

The number of swaps per sample is computationally less important, but the results again show that too low a number of swaps per sample in the serial burn-in sampling scheme can strongly decrease the quality of the result, even if $10,000$ samples are drawn from the random graph model. Fig. 2 shows the steep transitions when varying the number of swaps.

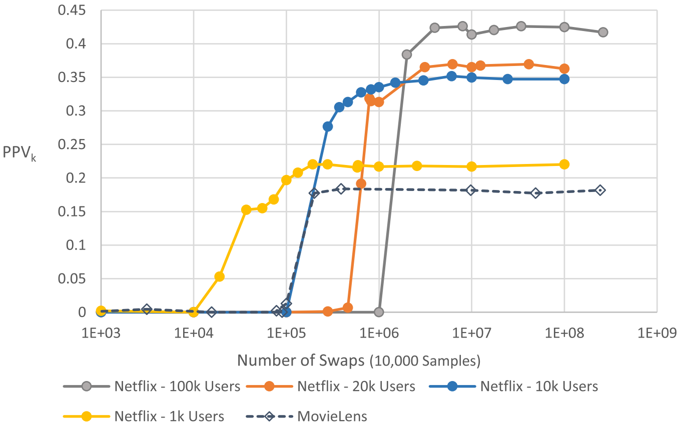

Fig. 2. Quality over number of swaps. For a wide variety of data sets, narrow phase transitions are present, similarly to what can be observed varying the number of samples.

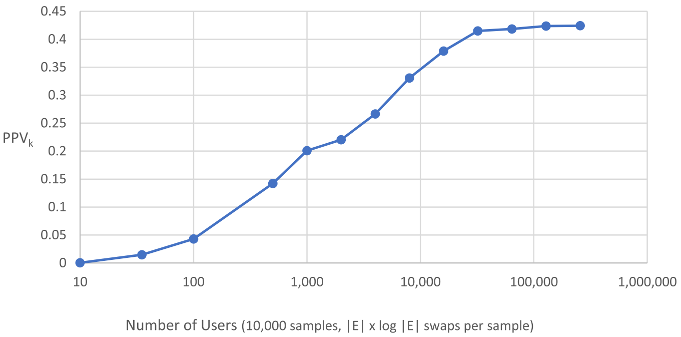

Fig. 3. Quality over the number of users in the Netflix data set. The users are selected at random from the 478k total users in the data set. Each point is the average of 10 random subsets.

Fig. 3 shows the quality of the ranking for different sizes of subsets of users in the bipartite graph. The maximally achievable quality strongly depends on the size of the set of users: with $1,000$ users, a PPV_k-value of only 0.20 ± 0.04 can be achieved, while the full data set allows for a PPV_k-value

[1] The 100k MovieLens data set, available from http://grouplens.org/datasets/movielens/.

of 0.424 ± 0.007. Fig. 3 shows the quality in dependence of the number of users and the number of samples from the random graph model. The data shows that taking $100,000$ users out of the $480,000$ uniformly at random (averaged over 10 of these sets) allows for the same overall quality, but requires more samples from the FDSM. It seems that, in general, a smaller set is sufficient if more samples from the random graph model are made to assess the significance of the observed co-occurrence values.

In this section we have shown that there are various phase transition-like changes in the resulting quality of the link assessment that should be considered when selecting a subset from a larger data set, the number of swaps in a serial burn-in-scheme, and the number of samples. While we had ground truth to evaluate this behavior in these cases, other data sets do not necessarily come with a precompiled ground truth. Thus, the next section introduces an online heuristic for large data sets that can be used to estimate the necessary number of swaps and samples to assess the significance of the observed co-occurrence in a bipartite graph.

V. Heuristics

Fig. 1 shows a sharp phase transition for the full Netflix data set from a PPV_k of 0 to 0.41 between 48 and 192 samples; after that the PPV_k is almost constant even up to 10k samples. Since the runtime is linear in the number of samples, there is no point in generating more than 192 samples from a practical perspective. However, we see in Fig. 1 that the transition is also data dependent. Analogously, the same observation can be made for the number of swaps. To reduce the overall runtime without decreasing the quality of the result, it is thus necessary to monitor the sampling process online and stop when the number of swaps and the number of samples is sufficient. In the following we propose two online heuristics that indicate the minimum required #samples and #swaps, without the usage of any kind of ground truth.

A. Heuristic for #swaps: Same Degree Coocc Convergence

The number of swaps needs to be high enough, otherwise the sampled graphs are not independent from the starting graph. The idea of the *swap heuristic* is to build a correlated variable θ that indicates when the Markov chain has mixed, i.e., the constructed graph is independent from the starting point. For small graphs, this number can be set to $m \log m$, the number of steps such that, expectedly, every edge has been selected at least once. For larger graphs, this number is prohibitively large. However, larger graphs are likely to contain a set of pairs of nodes with the same degree that start with very different co-occurrences. For example, there might be seven nodes with degree 10 and four nodes with degree 20. Thus, there are 28 pairs of nodes with the same degrees.

While in every random sample the $coocc(a, b)$ of two nodes a, b is different, we know that the average over all samples $coocc_{FDSM}(a, b)$ converges to a fixed number that only depends on the degrees of a and b. Thus, it is also the same for **all** pairs of nodes with the same degrees:

$$\begin{aligned} &\forall a, b, c, d \in V_L : deg(a) = deg(c) \wedge deg(b) = deg(d) \\ &\quad \Rightarrow coocc_{FDSM}(a, b) = coocc_{FDSM}(c, d). \qquad (1) \end{aligned}$$

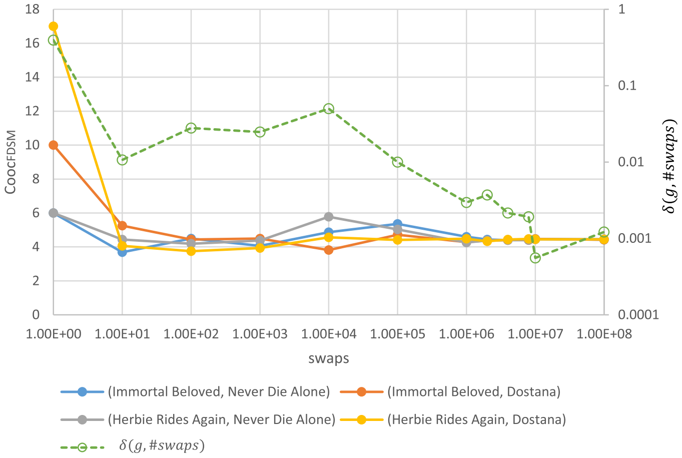

Fig. 4. Convergence of the $coocc_{FDSM}$ for the Netflix data set with 10k users.

Fig. 4 shows the $coocc_{FDSM}$ of four different movie pairs in the Netflix data, with the same degrees but different observed co-occurrences (points on the left of the x-axis). The figure shows the average co-occurrence of $10,000$ sampled graphs in dependence of the number of swaps in the serial burn-in sampling scheme. It can be clearly seen that the average co-occurrence of all pairs converges to the same value.

Based on this insight, we propose a function θ(#swaps) to determine the optimal number of swaps as described in the following: From the data set, extract all sets $D(x, y)$ of pairs of nodes that have at least N_p node pairs with the same degrees x and y. Compose G by selecting N_p pairs u.a.r. from all sets:

$$\begin{aligned} D(x, y) &= \{(a, b) \mid \forall a, b \in V_l : deg(a) = x, deg(b) = y\}, \\ G &= \{(d_1, .., d_{N_p}) \in D(x, y) \mid \forall x, y \in \mathbb{N} : |D(x, y)| \geq N_p\}. \end{aligned}$$

From this set G take N_g groups at random: $g_1, ..g_{N_g} \in G$. In each of these groups test the convergence of the average co-occurrence by computing the normalized standard deviation $\delta := s/m$ of the $coocc_{FDSM}$ in this group, where s is the standard deviation of the sample and m is its mean.

The function θ is then defined as the mean over all deviations δ, see Fig. 4.

$$\begin{aligned} \mathcal{C}(g_i, \#\text{swaps}) &= \{coocc^{\#\text{swaps}}_{FDSM}(a, b) \mid \forall (a, b) \in g_i\}, \\ \delta(g_i, \#\text{swaps}) &= \frac{std\left[\mathcal{C}(g_i, \#\text{swaps})\right]}{mean\left[\mathcal{C}(g_i, \#\text{swaps})\right]}, \\ \theta(\#\text{swaps}) &= \frac{1}{N_g} \sum_{i=1}^{N_g} \delta(g_i, \#\text{swaps}). \end{aligned}$$

To empirically assess the quality of the heuristic for finding the minimal number of required swaps, the ground truth can be used once again: Fig. 5 shows the value of θ over #swaps for $N_p = 4$ and $N_g = 24$ in blue and the PPV_k in yellow. The heuristic shows an almost inverted behavior with respect to the PPV_k, indicating a good correlation. Thus, when θ approaches 0, we assume that the quality of the resulting significance test is reliable.

Based on θ, we now identify a good #swap without relying on the ground truth. Note that for small #swaps it is very efficient to evaluate θ, even for 10k samples. Starting with a

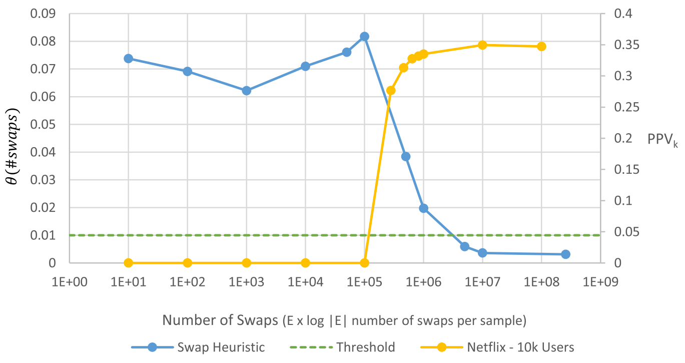

Fig. 5. Swap heuristic θ and the PPV.

low #swaps, we continuously increase it until $\theta < \theta_{min}$. A threshold of $\theta_{min} = 0.01$ appears to be a reasonable choice in our tests, which results in an average relative error of 1% in the *co-occurrence* (*coocc*). The complete swap heuristic is shown in Algorithm 1.

```
Data: Graph $G(V_L \cup V_R, E)$ with vertices $V_L$ and $V_R$
      and edges $E$, $V_R$ being the vertices of interest,
      $N_g, N_p, \theta_{min}$, #samples;
Result: #swaps
$G_0 := G$ randomized with $|E| \ln |E|$ swaps;
Select $N_g$ groups with $N_p$ pairs of nodes with same
degrees at random from $V_R$ each;
#swaps $:= \max(|V_L|, |V_R|) \ln \max(|V_L|, |V_R|)$;
$s_{high} := |E| \ln |E|$;
do
    #swaps $:= \sqrt{\text{\#swaps} \cdot s_{high}}$;
    Evaluate $\theta$(#swaps) with given #samples from $G_0$;
while $\theta(\#swaps) > \theta_{min}$;
```

Algorithm 1: Same Degree Coocc Convergence Swap Heuristic

B. Heuristic for #samples: Internal PPV_k

The heuristics for determining an ideal number of samples is based on the idea that the ranking of the most significant co-occurring pairs (pairs of nodes with the most significant number of common neighbors) should stabilize as the number of samples increases.

We thus propose to use the *internal PPV_k heuristic*, which makes use of an "internal ground truth", defined as the k pairs of nodes that were ranked highest in the previous iteration, where the ranking is based on the p-value and ties are broken with respect to the z-score. Then, based on this internal ground truth GT', the PPV_k of the current result is calculated, i.e., we quantify how much the newly sampled graph(s) change the ranking of the top k' pairs and stop if that value is larger than some threshold value α. Algorithm 2 shows the steps in details.

Constructing the internal ground truth GT', while not increasing the complexity of the algorithm, is still computationally relevant. Instead of constructing it for every sample, we therefore only construct it every $samples_{step}$ samples. The stopping quality α and the length k', in turn, should be high enough to guarantee a sufficient stability, while still being small enough to keep the overhead as small as possible.

```
Data: Graph $G(V_L \cup V_R, E)$ with vertices $V_L$ and $V_R$
      and edges $E$, $V_R$ being the vertices of interest,
      #swaps, $k'$, $samples_{step}$, $\alpha$;
Result: ranking according to $p$-value and $z$-score for all
        pairs of vertices $(u, v) \in (V_R \times V_R)$;
Calculate $coocc(u, v)\ \forall\ (u, v) \in (V_R \times V_R)$;
$G_0 := G$; $i := 0$;
do
    $GT' := k'$ pairs $(u, v)$ with the highest ranking of
    $G_i$ ;
    for $k$ := 1 to $samples_{step}$ do
        $i := i + 1$; $G_i := G_{i-1}$;
        Randomize $G_i$ with given #swap;
        Calculate $coocc_i(u, v)\ \forall\ (u, v) \in (V_R \times V_R)$;
    end
    $PPV' := PPV_{k'}$(List of $k'$ highest ranked node
    pairs containing at least one node from $GT', GT'$);
while $PPV' < \alpha$;
Calculate ranking according to Section V-A;
```

Algorithm 2: Internal PPV_k Sample Heuristic

In Fig. 6 the output of the heuristic for the full Netflix data set is shown, with the following configuration:

$$samples_{step} = 16; \quad k' = 0.2\% \ |V_R|^2; \quad \alpha = 0.95.$$

The PPV_k for the Movie Ground Truth is shown in blue. The dotted lines denote the internal PPV'. While the two curves for each data set are not perfectly aligned, the internal PPV_k converges later than the ground-truth-based PPV_k, such that it is safe to use the heuristic as a stopping point. Based on the graph, we conclude that the internal PPV_k gives us a stable and clear indicator of when to stop the algorithm.

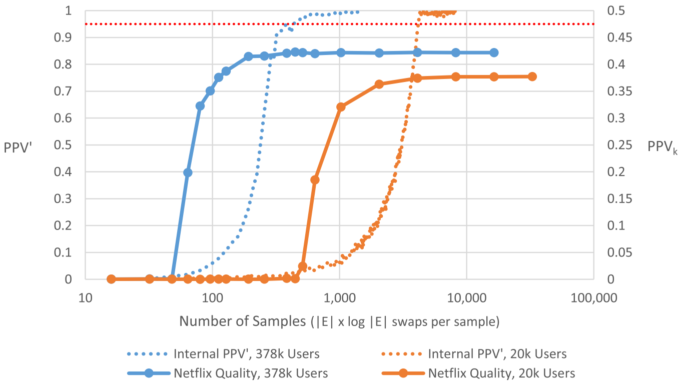

Fig. 6. Sample heuristic PPV' and PPV.

C. Results

Based on the two stopping heuristics, the runtime of the algorithm can be dramatically reduced, especially for the large Netflix data set (Table I): the swap heuristics alone appears to be helpful mostly for large enough graphs: the MovieLens data is so small that the overhead needed to compute the heuristics almost diminishes savings in the overall runtime. For the medium-sized Netflix subset (100k users), the heuristics decrease the runtime by a factor of 1.5. Computing the statistical significance of all co-occurrences with a "safe" burn-in phase of $m \log m = 10^9$ swaps and the same number of

TABLE I. RUNTIME AND QUALITY BEHAVIOUR OF THE HEURISTICS.

Data set	Samples	Swaps	Runtime	PPV
State-of-the-art				
Netflix, 487k users	10,000	10^9	20 h	0.422
Netflix, 100k users	10,000	2.6×10^8	5.5 h	0.425
MoviesLens	10,000	1.5×10^7	877 s	0.290
Swap Heuristic[1]:				
Netflix, 100k users	10,000	6.8×10^7 (**4x**)	0.2+3.4 h (1.5x)	0.424 (-0.2%)
MovieLens	10,000	1.5×10^6 (**10x**)	110+554 s (1.3x)	0.290 ($\pm 0.0\%$)
Sample Heuristic[2]:				
Netflix, 487k users	640	10^9	1.42 h (**14x**)	0.418 (-0.9%)
Netflix, 100k users	2,944	2.6×10^8	1.66 h (**3.3x**)	0.419 (-1.3%)
MovieLens	3,456	1.5×10^7	326 s (**2.7x**)	0.291 ($+1.0\%$)

[1] $N_p = 4$; $N_g = 24$; $\theta_{min} = 0.01$

[2] $samples_{step} = 128$; $k' = 0.2\% \, |V_R|^2$; $\alpha = 0.95$.

swaps for each subsequent sample with a total of $10,000$ samples took 20 hours. Mind that this time was only achievable by running 128 sampling chains in parallel on a cluster. For example, computing the algorithm on only one CPU would have taken approximately 3.5 months! While with the swap heuristic the computation time can be reduced by over an hour, the improvement of the runtime by reducing the number of samples from $10,000$ to 640 is even larger, namely a factor of 14x, while reducing the quality of the link assessment by only about 1%. Even if the algorithm ran on a single CPU, the runtime would now be only slightly over a day. Similarly, also the MovieLens data profits from the sample heuristics by reducing the runtime by a factor of 2.7.

VI. CONCLUSION

While the use of the FDSM to assess the statistical significance of structural patterns in graphs has often been proposed, its application was so far infeasible for large-scale data since there is no practically usable known upper bound on the mixing time of the Markov chain that allows to sample uniformly at random from the FDSM. Similarly, there is no known bound on the number of necessary samples to achieve a good quality of the *expected* values by the *observed* means in the sample.

Here we have shown that two online heuristics can help to determine a sufficient number of swaps and samples to achieve high-quality results with respect to the quality that is achievable with "safe" parameters. This discovery is based on the use of ground truth data that allows evaluating the quality of the resulting significance. The reduction in runtime by a factor of up to 14 on the used data makes it possible to apply the proposed algorithm to the full Netflix data set and get the result within two days on a single-core CPU without reducing the quality significantly. Further research will have to show whether the results can be transferred to other kind of network data like protein-protein interaction data or social network data. However, here it is much more difficult to obtain ground truth data for the evaluation.

Reducing the data to a much smaller subset without diminishing the method's capabilities is another important achievement and we showed that a random sample of 100k users from the Netflix data set is sufficient to obtain a result with almost the same quality as the much larger full data set with 480k users. Further research is necessary to understand whether there is a heuristic that does not rely on any ground truth to determine the necessary size of a subset of the data to achieve a high-quality link assessment.

ACKNOWLEDGMENT

We would like to thank Emőke-Ágnes Horvát for helpful discussions. The simulations were executed on the high performance cluster "Elwetritsch" at the TU Kaiserslautern which is part of the "Alliance of High Performance Computing Rheinland-Pfalz" (AHRP). We kindly acknowledge the support.

REFERENCES

[1] R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, and U. Alon, "Response to Comment on "Network motifs: Simple building blocks of complex networks " and "Superfamilies of evolved and designed networks"," *Science*, vol. 305, p. 1107d, 2004.

[2] K. A. Zweig, "How to forget the second side of the story: A new method for the one-mode projection of bipartite graphs," in *Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining ASONAM 2010*, 2010, pp. 200–207.

[3] K. A. Zweig, "Good versus optimal: Why network analytic methods need more systematic evaluation," *Central European Journal of Computer Science*, vol. 1, pp. 137–153, 2011.

[4] S. Uhlmann, H. Mannsperger, J. D. Zhang, E.-Á. Horvat, C. Schmidt, M. Küblbeck, A. Ward, U. Tschulena, K. Zweig, U. Korf, S. Wiemann, and Ö. Sahin, "Global miRNA regulation of a local protein network: Case study with the EGFR-driven cell cycle network in breast cancer," *Molecular Systems Biology*, vol. 8, p. 570, 2012.

[5] E.-Á. Horvát, J. D. Zhang, S. Uhlmann, Ö. Sahin, and K. A. Zweig, "A network-based method to assess the statistical significance of mild co-regulation effects," *PLOS ONE*, vol. 8, no. 9, p. e73413, 2013.

[6] E.-Á. Horvát and K. A. Zweig, "A fixed degree sequence model for the one-mode projection of multiplex bipartite graphs," *Social Network Analysis and Mining*, vol. 4, p. 164, 2013.

[7] C. I. Del Genio, H. Kim, Z. Toroczkai, and K. E. Bassler, "Efficient and exact sampling of simple graphs with given arbitrary degree sequence," *PloS one*, vol. 5, no. 4, p. e10012, 2010.

[8] R. Kannan, P. Tetali, and S. Vempala, "Simple markov-chain algorithms for generating bipartite graphs and tournaments," *Random Structures and Algorithms*, vol. 14, no. 4, pp. 293–308, 1999.

[9] A. Berger and M. Müller-Hannemann, "Uniform sampling of digraphs with a fixed degree sequence," in *Graph theoretic concepts in computer science*, 2010, pp. 220–231.

[10] G. W. Cobb and Y.-P. Chen, "An application of Markov Chain Monte Carlo to community ecology," *The American Mathematical Monthly*, vol. 110, pp. 265–288, 2003.

[11] M. Jerrum, A. Sinclair, and E. Vigoda, "A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries," *Journal of the ACM (JACM)*, vol. 51, no. 4, pp. 671–697, 2004.

[12] R. A. Brualdi, "Matrices of zeros and ones with fixed row and column sum vectors," *Linear Algebra Applied*, vol. 33, pp. 159–231, 1980.

[13] A. Gionis, et al., "Assessing data mining results via swap randomization," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 3, p. article no. 14, 2007.

[14] A. Gionis, H. Mannila, T. Mielikäinen, and P. Tsaparas, "Assessing data mining results via swap randomization," in *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06)*, 2006.

[15] E.-Á. Horvát and K. A. Zweig, "One-mode projections of multiplex bipartite graphs," in *Proceedings of the 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, 2012.

[16] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, May 2007. [Online]. Available: http://doi.wiley.com/10.1002/asi.20591