

PHENOMENA IN GROWING NETWORKS AND LEARNING
ACROSS NETWORKS

Von der Fakultät für Informatik, Elektrotechnik und Informationstechnik der
Universität Stuttgart zur Erlangung der Würde eines Doktors der
Naturwissenschaften (Dr. rer. nat.) genehmigte Abhandlung

Vorgelegt von

JUN SUN

aus Nanjing, China

Hauptberichter: Prof. Dr. Steffen Staab
Mitberichter: Prof. Dr. Ciro Cattuto

Tag der mündlichen Prüfung: 02.02.2022

Institut für Parallele und Verteilte Systeme (IPVS) der Universität Stuttgart

2022

ABSTRACT

In this dissertation, I study two topics in network science, namely, phenomena in growing networks, and machine learning across networks.

The first topic of the dissertation includes the empirical study of different phenomena in real networks at both the microscopic and the macroscopic level, as well as the analytical study of preferential attachment network models. I review existing preferential attachment network models, and give remarks on what exactly the models explain and what can emerge from the models in terms of real-world phenomena at different levels. In the empirical study that follows, I particularly look into two citation networks, and find that (1) the network size grows exponentially over time; (2) the degree growth in citation networks is time-invariant. Existing preferential attachment models cannot explain these two additional phenomena at the same time. I propose a novel analytical framework for a general set of preferential attachment network models, where I connect the growth of the network size and the growth of the node degrees as an eigenproblem. I show that there exist only two solutions to the eigenproblem: Network size growing linearly or exponentially with time. The sometimes lack of solution to the eigenproblem corresponds to the breaking of the system's time-invariance, which explains the winner-takes-all effect in some model settings, revealing the connection between the Bose-Einstein condensation in the Bianconi-Barabási model and a similar condensation in superlinear preferential attachment. I also show how to estimate the network properties using our framework, for instance the degree distribution and the exponential growth rate of the network. I prove that the ageing effect is necessary to reproduce realistic node degree growth curves, and can prevent the winner-takes-all effect under weak conditions. At last, I use extensive numerical simulations to verify our analytical results, and show that the generated networks have realistic scale-free degree distributions depending on the parameters.

In the second topic of the dissertation, I investigate how machine learning techniques, in particular transfer learning, can be applied in real-world networks. Akin to human transfer of experiences from one domain to the next, transfer learning as a subfield of machine learning adapts knowledge acquired in one domain to a new domain. I systematically investigate how the concept of transfer learning may be applied to the study of users on Web platforms, and propose our transfer learning approach, TraNet. TraNet is based on feature transformation from each network's local feature distribution to a global feature space. I explain our approach in detail with the experiments on the user interaction networks of Wikipedia in different languages. I then show how TraNet is applied to tasks involving the identification of trusted users on Web platforms. I compare the performance of TraNet with other approaches and find that our approach can best transfer knowledge on users across platforms in the given tasks.

ZUSAMMENFASSUNG

In dieser Dissertation untersuche ich zwei Themen der Netzwerkwissenschaft, nämlich Phänomene in wachsenden Netzwerken und netzwerkübergreifendes maschinelles Lernen.

Das erste Thema der Dissertation umfasst die empirische Untersuchung verschiedener Phänomene in realen Netzwerken sowohl auf mikroskopischer als auch auf makroskopischer Ebene, sowie die analytische Untersuchung von Preferential Attachment Netzwerkmodellen. Ich überprüfe die bestehende Preferential Attachment Netzwerkmodelle, und gebe Anmerkungen dazu, was genau die Modelle erklären und was sich aus den Modellen in Bezug auf Phänomene der realen Welt auf verschiedenen Ebenen ergeben kann. In der folgenden empirischen Studie untersuche ich insbesondere zwei Zitationsnetzwerke und stelle fest, dass (1) die Netzwerkgröße mit der Zeit exponentiell wächst; (2) der Gradwachstum in Zitationsnetzwerken zeitinvariant ist. Bestehende Preferential Attachment Modelle können diese beiden zusätzlichen Phänomene nicht gleichzeitig erklären. Ich schlage einen neuartigen analytischen Rahmen für einen allgemeinen Satz von Preferential Attachment Netzwerkmodellen vor, wobei ich das Wachstum der Netzwerkgröße und das Gradwachstum als Eigenproblem verbinde. Ich zeige, dass es nur zwei Lösungen für das Eigenproblem gibt: Netzwerkgröße wächst linear oder exponentiell mit der Zeit. Die manchmal fehlende Lösung des Eigenproblems korrespondiert mit dem Aufbrechen der Zeitinvarianz des Systems, was den „Winner-takes-all“ Effekt in einigen Modellsettings erklärt und den Zusammenhang zwischen der Bose-Einstein-Kondensation im Bianconi-Barabási-Modell und einer ähnlichen Kondensation in superlinearer Preferential Attachment aufdeckt. Ich zeige auch, wie man die Netzwerkeigenschaften mit unserem Rahmenwerk schätzen kann, zum Beispiel die Gradverteilung und die exponentielle Wachstumsrate des Netzwerks. Ich beweise, dass der Alterungseffekt notwendig ist, um realistische Gradwachstumskurven zu reproduzieren, und dass er den „Winner-takes-all“ Effekt unter schwachen Bedingungen verhindern kann. Zuletzt verwende ich umfangreiche numerische Simulationen, um unsere analytischen Ergebnisse zu verifizieren, und zeige, dass die generierten Netzwerke realistische skalenfreie Gradverteilungen in Abhängigkeit von den Parametern haben.

Im zweiten Thema der Dissertation untersuche ich, wie Techniken des maschinellen Lernens, insbesondere Transfer-Lernens, in Netzwerken der realen Welt angewendet werden können. Ähnlich dem menschlichen Transfer von Erfahrungen von einer Domäne in die nächste, passt Transfer-Lernen als Teilbereich des maschinellen Lernens das in einer Domäne erworbene Wissen an eine neue Domäne an. Ich untersuche systematisch, wie das Konzept des Transfer-Lernens auf die Untersuchung von Benutzern auf Web-Plattformen angewandt werden kann, und schlage unseren Transfer-Lernansatz TraNet vor. TraNet basiert auf der Merkmalstransformation von

der lokalen Merkmalsverteilung jedes Netzwerks in einen globalen Merkmalsraum. Ich erläutere unseren Ansatz im Detail anhand der Experimente zu den Benutzerinteraktionsnetzwerken von Wikipedias in verschiedenen Sprachen. Anschließend zeige ich, wie TraNet auf Aufgaben angewendet wird, bei denen es um die Identifizierung von vertrauenswürdigen Benutzern auf Web-Plattformen geht. Ich vergleiche die Leistung von TraNet mit anderen Ansätzen und stelle fest, dass unser Ansatz das Wissen über Benutzer in den gegebenen Aufgaben am besten plattformübergreifend übertragen kann.

ACKNOWLEDGMENTS

I could not have written this dissertation without the help and support from many people.

I would like to first thank my supervisor, Professor Steffen Staab, for invaluable guidance and inspiration during my entire PhD journey.

I would like to thank Jérôme Kunegis for introducing me into the world of network science.

I would like to thank all co-authors of the publications during my PhD, Jérôme Kunegis, Steffen Staab, Fariba Karimi, Matúš Medo, Korok Sengupta, Raphael Menges, Chandan Kumar, for the great time working together.

I would like to thank my colleague Sarah de Nigris for the helpful discussion on the topic of network models. I would like to thank Albert-László Barabási for clearing my doubt about the Bose-Einstein condensation in networks.

The research leading to the results in this dissertation has received funding from the European Community's Seventh Framework Programme project REVEAL¹ under grant agreement No. 610928, and the EU Horizon 2020 project CUTLER² under contract No. 770469. I would like to thank Jérôme Kunegis, Chandan Kumar and other colleagues of mine for the tacit collaboration in the projects.

I would like to thank everyone else that I have not mentioned and that have discussed with me at conferences, workshops and seminars. I would like to thank all reviewers for taking the time and effort to review my publications and give feedback.

I would like to thank the American Physical Society (APS) for providing the APS citation data. I would like to thank Software AG, our partner in the REVEAL project, for providing the ARIS Community dataset. I would like to thank the KONECT project³ for hosting multiple network datasets.

Finally, I would like to thank my wife Yuan Sun, my little daughter An, and all my family for the encouragement, and for always being my support.

¹ <https://revealproject.eu>

² <https://www.cutler-h2020.eu>

³ <http://konect.cc>

CONTENTS

1	INTRODUCTION	1
1.1	Background and motivation: Phenomena in growing networks	1
1.2	Background and motivation: Learning across networks	2
1.3	Outline	4
1.4	Publications, talks and own contributions	5
2	FOUNDATIONS	7
2.1	Mathematics	7
2.1.1	Matrices and vectors	7
2.1.2	Matrix multiplication as linear transformation	10
2.1.3	Eigenvalue problems	10
2.1.4	Laplace transform	13
2.1.5	Linear time-invariant system	15
2.2	Graphs and networks	17
2.2.1	Network types	18
2.2.2	Common terminology	19
2.3	Matrix representations of networks	21
2.3.1	Adjacency matrix	21
2.3.2	Degree matrix	22
2.4	Degree distribution	23
2.4.1	Degree distribution of the Erdős-Rényi model	24
2.4.2	Degree distribution of real-world networks	24
2.4.3	Continuous approximation of the degree distribution	25
2.5	Structural properties	27
2.5.1	Degree centrality	27
2.5.2	Eigenvector centrality	27
2.5.3	Random walk centrality	28
2.5.4	PageRank	29
2.5.5	Coreness and k -core	30
2.5.6	Clustering coefficients	31
3	PHENOMENA IN GROWING NETWORKS	33
3.1	Related work	34
3.1.1	Preferential attachment models	34
3.1.2	Mean-field estimation of the degree distribution	36
3.1.3	Bianconi-Barabási model	39
3.1.4	Bose-Einstein condensation in networks	42
3.1.5	Relevance decay model	48
3.2	Empirical observations	49
3.2.1	Datasets	51
3.2.2	Time-invariant degree growth	53
3.2.3	Varying average outdegree over time	53
3.2.4	Exponential network growth	56

3.2.5	Indegree distribution	56
3.3	General preferential attachment network model	58
3.3.1	Problems in existing models	58
3.3.2	General preferential attachment with fitness, ageing and exponential network growth	60
3.3.3	Ageing functions	62
3.4	A novel formalism of preferential attachment models	65
3.4.1	Estimation of the degree distribution	67
3.4.2	Relation to existing preferential attachment models	68
3.5	Breaking of time-invariance	79
3.5.1	Breaking of time-invariance in Bianconi-Barabási model	80
3.5.2	Breaking of time-invariance in superlinear preferential attachment	81
3.5.3	No condensation in the presence of ageing	81
3.6	Evaluation	84
3.6.1	Methodology	84
3.6.2	Evaluation focusses	85
3.6.3	Results	87
3.6.4	Result analysis	101
3.7	Summary	103
4	LEARNING ACROSS NETWORKS	105
4.1	Related work	106
4.1.1	Transfer learning	106
4.1.2	Transductive transfer learning	107
4.1.3	Transfer learning algorithms for network analysis	107
4.2	Proposed method	108
4.2.1	Overview	111
4.2.2	Base feature extraction	112
4.2.3	Feature transformation	117
4.2.4	Feature aggregation	121
4.2.5	Reduction to classification problem	125
4.3	Evaluation	125
4.3.1	Evaluation metric	126
4.3.2	Baselines	126
4.3.3	Evaluation of role transfer in Wiki-talk networks	127
4.3.4	Evaluation of trust transfer in signed networks	128
4.4	Summary	130
5	CONCLUSION	131
5.1	Conclusion and outlook: Phenomena in growing networks	131
5.2	Conclusion and outlook: Learning across networks	132
A	APPENDIX: SUPPLEMENTARY RESULT FOR CHAPTER 3	133
B	APPENDIX: SUPPLEMENTARY RESULT FOR CHAPTER 4	157
C	APPENDIX: WIKI-TALK DATASET	237
	NOMENCLATURE	241
	LIST OF FIGURES	243
	LIST OF TABLES	249
	BIBLIOGRAPHY	251

INTRODUCTION

Network science is a highly interdisciplinary area in terms of both its theoretical foundation and practical application due to the ubiquity of network structures in many fields of study. To understand the relations between one and another of these fields is both beneficial and challenging. The dissertation of mine covers two major topics in network science, namely, generative network models and machine learning across networks, respectively from two different fields.

1.1 BACKGROUND AND MOTIVATION: PHENOMENA IN GROWING NETWORKS

Growing network models investigate the underlying mechanics of how networks are formed from preferably simple rules, leading to better understandings of how different phenomena in real networks can emerge. The original work on the preferential attachment network growth model [5] has importantly contributed to the formation of the interdisciplinary field of network science [7, 55]. The basic preferential attachment mechanism, also known as the rich-get-richer or the Matthews effect, dictates that nodes attract new links at a rate that is proportional to the degree that they already have. This microscopic mechanism induces a positive feedback loop that results in a network degree distribution that is power-law (scale-free) under some model settings [3]. Similar broad degree distributions, though seldom of an ideal power-law shape, are found in many real-world networks [18, 52]. Since then, preferential attachment-based network models have been used to model the evolution of a broad range of networks, such as the World Wide Web [1, 5], citation networks [35, 50], and social networks [16, 52]. The most important generalisations of the original preferential attachment model are the inclusion of the node-specific fitness parameter [10], and ageing that suppresses the attractiveness of old nodes to new links [22].

What motivates our study in this topic is our empirical observation, particularly from two citation networks, that (1) the network size grows exponentially over time; (2) the degree growth in citation networks

is time-invariant, *i.e.*, the average number of citations received by a paper after τ years is independent of the paper's publication time. When reviewing existing preferential attachment network models, we analyse what exactly the models explain and what can emerge from the models in terms of real-world phenomena at both microscopic and macroscopic levels. We argue that existing preferential attachment models cannot explain the observed phenomena at the same time.

Moreover, different network condensation phenomena have been studied in literatures, for instance in the Bose-Einstein condensation in the Bianconi-Barabási model with skewed fitness distributions [9], and in the superlinear preferential attachment model [40]. The condensation in networks corresponds to the "winner-takes-all" effect in the real world, as an extreme case of the rich-get-richer effect. However, the connection between the different network condensation phenomena still remains unrevealed [9].

Thus, we aim at proposing a novel analytical framework for a general set of preferential attachment network models, that can connect the growth of the network size and the growth of the node degrees, and can clarify the difference and connection between the linear network growth which is usually assumed by existing preferential models, and the exponential network growth which is often observed in reality. Within our framework, we also aim at unifying the different network condensation phenomena that have been studied separately, and studying under what circumstances we can prevent the network from undergoing a condensation.

1.2 BACKGROUND AND MOTIVATION: LEARNING ACROSS NETWORKS

Network is a powerful tool not only to study citation and growth patterns as we have mentioned in the first topic. It can also model many other real systems, such as communities of people. Individual actors (nodes) in the communities and the interactions (links) among them are often modelled as different types of social networks. A detailed understanding of the actors in the network can contribute to the understanding of the whole community. For example, individual actors can have different *labels* in social networks: Some nodes are people bridging two sub-communities, some are central people through which a large part of communication passes, some outliers. Other examples of labels can be, for instance, "trolls" or users' trustworthiness.

For large online social networks, the only scalable way to annotate all users with labels is through automatic labelling of nodes, *i.e.*, using machine learning to train a model from existing, annotated data, and infer the labels of unknown nodes. However, when studying a completely new network where no labels are known, we have to rely on our knowledge about existing networks, because the labels cannot be inferred from the network itself. An unsupervised learning algorithm could be used, but it would only output unlabelled classes without meaning attached to the classes.

The main challenge of learning from existing networks lies in the fact that different communities are so heterogeneous in terms of size and structure. As a simple example, a user with 20 friends in a small-scale network (*e.g.*, a friendship network in a classroom) might be considered influential, while a user with the same number of friends in Facebook is far from being influential. Some platforms such as Slashdot contain negative user relationships such as “foes”, and others do not.

Quantifiable cross-community analyses have been undertaken by, *e.g.*, Rowe et al. [64]. They have proposed several measures useful to quantify differences between communities, such as the number of initiative-takers or the length of discussions. Considering these measures, however, it still remains open whether and how they could actually be re-used to transfer experiences from one community to another. The learning process would be cut short greatly, if measurements of social behaviour could be transferred from previous experiences to new ones, not just based on qualitative observations, but also based on quantifiable rules. For example, a new Web platform might want to discourage trolls and encourage trusted users without running through the learning cycle multiple times by transferring quantitative experiences from previous Web platforms.

As the second topic of the dissertation, we aim at investigating how *transfer learning* can be applied in real-world networks. Akin to human transfer of experiences from one domain to the next, transfer learning, as a subfield of machine learning, enables one to learn knowledge from other labelled datasets, and apply the learnt knowledge to the unlabelled dataset [61]. In transfer learning approaches, an algorithm is trained on one dataset (the source dataset) and applied to another dataset (the target dataset). In practice, this only works as stated if the two datasets have very similar structural (and other) properties. Thus, we also need to additionally investigate different functions that effectively transfer features from one dataset to another, or to a common feature space. Such transfer is particularly challenging due to the above-mentioned heterogeneity of Web platforms and the

sometimes drastic differences in structural network features such as the degree distribution.

1.3 OUTLINE

In this section we outline the overall structure of the remaining dissertation.

CHAPTER 2 introduces the foundational concepts that will be used throughout this dissertation, including the basic mathematics that is needed to follow the derivations in the dissertation, and the notational conventions.

CHAPTER 3 introduces the observation of various phenomena in growing networks, and our work on preferential attachment network models that try to explain the phenomena. This chapter makes the following contributions:

- We systematically review various existing preferential attachment network models, and analyse how they each explains various phenomena in real systems such as growing networks.
- We empirically observe the exponential network growth and the time-invariance of the degree growth in real networks.
- We propose a novel analytical framework for preferential attachment network models based on the observations.
- We reveal the connection between different condensation effects using our new framework.

CHAPTER 4 introduces our work on machine learning techniques for cross-network learning. This chapter makes the following contributions.

- We propose our transfer learning approach, TraNet, for cross-network learning tasks such as user studies in online platforms.
- We propose a method of feature transformation for power-law distributions, which can be used effec-

tively in cross-network learning tasks on features such as node degrees.

- We evaluate TraNet with concrete use cases: Classifying users into different roles study the user interaction networks of Wikipedia in different languages, and identifying trusted users in ARIS Community.

CHAPTER 5 concludes the whole dissertation.

In addition, Appendix A provides the supplementary experimental results for Chapter 3, and Appendix B provides the supplementary experimental results for Chapter 4. Appendix C describes the Wikitalk datasets in detail, and illustrates how to extract the datasets in order to reproduce the results in Chapter 4.

1.4 PUBLICATIONS, TALKS AND OWN CONTRIBUTIONS

This dissertation contains material in papers that were published already in conference proceedings and journals as listed below [69–72]. Papers that are not relevant to this dissertation are omitted.

- Time-invariant degree growth in preferential attachment network models
Sun, J., Medo M., & Staab S. In: Physical Review E 101.2 (2020).
- Decay of Relevance in Exponentially Growing Networks
Sun, J., Steffen S., & Fariba K. In: Proceedings of the 10th ACM Conference on Web Science (2018).
- Understanding Social Networks using Transfer Learning
Sun, J., Staab, S., & Kunegis, In: Computer 51.6 (2018).
- Predicting User Roles in Social Networks using Transfer Learning with Feature Transformation
Sun, J., Kunegis, J., & Staab, S. In: Proceedings of the IEEE 16th International Conference on Data Mining Workshops (2016).

In addition, I participated the following conferences as a speaker for regular talks. Contributions to these conferences include peer reviewed conference or workshop papers (see above) and abstracts with figures. This dissertation contains material in the presentations of these talks. Talks that are not relevant to this dissertation are omitted.

- IEEE, 16th International Conference on Data Mining (ICDM 2016), Regular talk (Workshop), Barcelona.
- ACM, 10th Conference on Web Science (WebSci 2018), Regular talk, Amsterdam.
- Network Science Society, International School and Conference on Network Science (NetSci 2018), Regular talk, Paris.
- Network Science Society, International School and Conference on Network Science (NetSci 2020), Regular talk, Rome (Online).

I am the main author of the publications, and the main contributor of the talks listed above. Material included from these publications and presentations was authored by myself.

FOUNDATIONS

In this chapter, we introduce the foundational concepts that will be used in this dissertation. First, we illustrate the mathematics that is needed to follow the calculations in the dissertation, and define the notational conventions in Section 2.1. Then, we introduce concepts on graphs and networks, including the basic definitions of graphs and networks in Section 2.2, matrix representations of networks in Section 2.3, network degree distributions in Section 2.4, and structural properties in graphs in Section 2.5.

2.1 MATHEMATICS

To make sure that the text clearly reflects our thoughts and avoid ambiguity as much as possible, very often we use the language of mathematics to explain the ideas, although some of the derivation might seem a little bit hand waiving in the eyes of the most rigorous mathematicians. Mathematics also helps deriving new knowledge that is otherwise difficult to derive. In this section, very briefly, we illustrate the necessary mathematics that is needed to follow the calculations in the dissertation, and define the notational conventions that will be used consistently.

2.1.1 *Matrices and vectors*

Matrices and vectors are widely used to represent different kinds of information. They can also conveniently express linear transformation. In this dissertation in particular, they will be used when discussing both topics: Phenomena in growing networks, and machine learning across networks. In this subsection, we introduce basic notions of matrices and vectors.

Matrices

A matrix is essentially a table of numbers arranged in rows and columns. For instance,

$$\mathbf{X} = \begin{pmatrix} 1 & -1 & 2 \\ -1 & 2 & 0 \end{pmatrix} \quad (2.1)$$

We will use a bold, upper-case Latin letter to denote a matrix, for instance, \mathbf{A} , \mathbf{X} , \mathbf{Y} , etc.

is a matrix of 2×3 real values. Without explicitly mentioning, a matrix will contain only real numbers in this dissertation, *i.e.*, a matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ of size $m \times n$ consists of mn real numbers arranged with the indices $i \in \{1, \dots, m\}$ and $j \in \{1, \dots, n\}$, where “ $m \times n$ ” is called the *dimension* of the matrix. The element in the i th row and j th column of \mathbf{X} is denoted as $X_{i,j}$. The i th row of \mathbf{X} is denoted as $\mathbf{X}_{i,:}$. The j th column of \mathbf{X} denoted as $\mathbf{X}_{:,j}$.

A matrix is called a *square matrix* if $m = n$. The *transpose* of a square matrix \mathbf{X} , denoted as \mathbf{X}^T , is defined via

$$(\mathbf{X}^T)_{i,j} = X_{j,i}. \quad (2.2)$$

A square matrix \mathbf{X} is called a *symmetric matrix* if $\mathbf{X} = \mathbf{X}^T$.

A square matrix \mathbf{X} is called a *diagonal matrix* if its off-diagonal elements are all zeros, *i.e.*, $X_{i,j} = 0$ whenever $i \neq j$.

A diagonal matrix \mathbf{I} is called an *identity matrix* if its diagonal elements are all ones, *i.e.*, $I_{i,j} = \delta(i, j)$, where $\delta(i, j)$ is the Kronecker delta function. \mathbf{I}_n denotes the $n \times n$ identity matrix.

The *product* of two matrices $\mathbf{X} \in \mathbb{R}^{m \times l}$ and $\mathbf{Y} \in \mathbb{R}^{l \times n}$ is an $m \times n$ matrix \mathbf{XY} , whose elements are defined via

$$(\mathbf{XY})_{i,j} = \sum_{k=1}^l X_{i,k} Y_{k,j}. \quad (2.3)$$

The *inverse* of a square matrix $\mathbf{X} \in \mathbb{R}^{n \times n}$, denoted as \mathbf{X}^{-1} , is defined via

$$\mathbf{X}^{-1}\mathbf{X} = \mathbf{X}\mathbf{X}^{-1} = \mathbf{I}_n \quad (2.4)$$

Note not all square matrix have an inverse.

Vectors

A vector is a list of numbers, arranged in a row or column. For instance,

$$\mathbf{x} = [1 \quad -1 \quad 2] \quad (2.5)$$

$$\mathbf{y} = \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix} \quad (2.6)$$

are a row vector of size 3, and a column vector of size 3, respectively. Without explicitly mentioning, a vector will contain only real numbers in this dissertation, *i.e.*, a vector $\mathbf{x} \in \mathbb{R}^n$ of size n consists of n real numbers, arranged with the indices $i \in \{1, \dots, n\}$. The i th element in \mathbf{x} is denoted as x_i .

We will use a bold, lower-case Latin letter to denote a vector, for instance, \mathbf{v} , \mathbf{x} , \mathbf{y} , etc.

A column vector of size m can be seen as an $m \times 1$ matrix. A row vector of size n can be seen as a $1 \times n$ matrix. Thus, in the example given above, the row vector \mathbf{x} and the column vector \mathbf{y} can be seen as the transpose of each other, *i.e.*, $\mathbf{x} = \mathbf{y}^T$ and $\mathbf{y} = \mathbf{x}^T$.

The *dot product* of two real vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ is defined as

$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^n x_i y_i. \quad (2.7)$$

If $\mathbf{x} \cdot \mathbf{y} = 0$, the two vectors \mathbf{x} and \mathbf{y} are said to be *orthogonal*.

The *norm* of a real vector $\mathbf{x} \in \mathbb{R}^n$, denoted as $\|\mathbf{x}\|$, is defined as the square root of the dot product of \mathbf{x} and itself, *i.e.*,

$$\|\mathbf{x}\| = \sqrt{\mathbf{x} \cdot \mathbf{x}} = \sqrt{\sum_{i=1}^n x_i^2}. \quad (2.8)$$

More strictly speaking this is the vector's L^2 norm, which is also the vector's Euclidean length.

If $\|\mathbf{x}\| = 1$, the vector \mathbf{x} is called a *unit vector*.

A unit vector has length 1.

2.1.2 Matrix multiplication as linear transformation

Now we look into a special case of matrix multiplication. Let \mathbf{M} be an $m \times n$ matrix, and \mathbf{v} be a column vector of size n . Now we multiply the matrix \mathbf{M} with the vector \mathbf{v} :

$$\mathbf{w} = \mathbf{M}\mathbf{v}. \quad (2.9)$$

The resulting vector \mathbf{w} is a column vector of size m . The multiplication $\mathbf{M}\mathbf{v}$ itself can be seen as applying a *transformation*, represented by the matrix \mathbf{M} , to the column vector \mathbf{v} . In other words, the matrix \mathbf{M} defines a mapping from \mathbb{R}^n to \mathbb{R}^m .

Moreover, the transformation represented by matrix multiplication is *linear*. Let \mathbf{u} be a column vector of the same size as \mathbf{v} . Let c be any real valued scalar. By using the definition of the matrix multiplication we can get

$$\mathbf{M}\mathbf{v} + \mathbf{M}\mathbf{u} = \mathbf{M}(\mathbf{v} + \mathbf{u}), \quad (2.10)$$

$$c\mathbf{M}\mathbf{v} = \mathbf{M}(c\mathbf{v}). \quad (2.11)$$

The two equations above show that the transformation represented by matrix multiplication fulfils the two properties of linearity: *additivity* and *homogeneity of degree one*.

- Equation (2.10) shows that the transformation is additive: The order of the addition and the matrix multiplication does not affect the end result.
- Equation (2.11) shows that the transformation is homogeneous of degree one: The order of the constant scaling and the matrix multiplication does not affect the end result.

Thus we say that the transformation represented by matrix multiplication is a linear transformation.

2.1.3 Eigenvalue problems

Let \mathbf{M} be an $n \times n$ matrix, and \mathbf{v} be a column vector of size n . If

$$\mathbf{M}\mathbf{v} = \lambda\mathbf{v}, \quad (2.12)$$

Linearity of matrix
multiplication

the vector \mathbf{v} is called a *right eigenvector* of the matrix \mathbf{M} , and λ is called the corresponding *eigenvalue*. Intuitively, applying the linear transformation \mathbf{M} upon the eigenvector \mathbf{v} does not rotate the direction of \mathbf{v} , but merely scales \mathbf{v} by a factor of λ , and in case $\lambda < 0$, the direction of \mathbf{v} is reversed.

Similarly, if \mathbf{v} is a row vector of size n , and

$$\mathbf{v}\mathbf{M} = \lambda\mathbf{v}, \quad (2.13)$$

the vector \mathbf{v} is called a *left eigenvector* of the matrix \mathbf{M} , and λ is called the corresponding *eigenvalue*. However, without explicitly mentioning, eigenvectors will always refer to right eigenvectors in this dissertation.

2.1.3.1 Eigendecomposition

From Equation (2.13) one can see that, any eigenvalue of the matrix \mathbf{M} must fulfil the following characteristic equation:

$$\det(\mathbf{M} - \lambda\mathbf{I}) = 0, \quad (2.14)$$

where $\det()$ represents the matrix *determinant*. Equation (2.14) is an n th order polynomial equation which has n solutions for the eigenvalue λ , denoted as λ_i where $i = 1, 2, \dots, n$. The set of all eigenvalues λ is called the *spectrum* of \mathbf{M} . The spectrum can be arranged in the diagonal eigenvalue matrix $\mathbf{\Lambda}$ such that

$$\mathbf{\Lambda}_{i,i} = \lambda_i \quad (2.15)$$

with

$$|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n|. \quad (2.16)$$

If the $n \times n$ matrix \mathbf{M} has n linearly independent eigenvectors $\mathbf{v}_{(i)}$ corresponding to the eigenvalues λ_i where $i = 1, 2, \dots, n$, they can be arranged in the eigenvector matrix \mathbf{Q} such that the i th column of \mathbf{Q} is the i th unit eigenvector $\mathbf{v}_{(i)}$, i.e.,

$$\mathbf{Q}_{:,i} = \mathbf{v}_{(i)} \quad (2.17)$$

with

$$\|\mathbf{v}_{(i)}\| = 1. \quad (2.18)$$

Here we use the notation $\mathbf{v}_{(i)}$ in order to differentiate with \mathbf{v}_i , the i th element in the vector \mathbf{v} .

The matrix \mathbf{M} can be decomposed as the product of three matrices, *i.e.*,

$$\mathbf{M} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}. \quad (2.19)$$

This is called the *eigendecomposition* of \mathbf{M} .

In addition, if \mathbf{M} is a real symmetric matrix, the eigenvector matrix \mathbf{Q} can be arranged to be an *orthonormal matrix*, *i.e.*,

$$\mathbf{U}\mathbf{U}^T = \mathbf{I}. \quad (2.20)$$

“Orthonormal” means both orthogonal and normal.

2.1.3.2 Operators and eigenfunctions

We have shown that, if a vector \mathbf{y} of size n equals the multiplication of an $n \times n$ matrix \mathbf{A} and a vector \mathbf{x} of size n , *i.e.*,

$$\mathbf{y} = \mathbf{A}\mathbf{x}, \quad (2.21)$$

then \mathbf{y} can be seen as the result of performing a linear transformation represented by the matrix \mathbf{A} upon the vector \mathbf{x} . Particularly, if $\mathbf{y} = \lambda\mathbf{x}$, with λ being a scalar value, both \mathbf{x} and \mathbf{y} are eigenvectors of the matrix \mathbf{A} with the corresponding eigenvalue λ . In other words, the direction of the vectors \mathbf{x} and \mathbf{y} is invariant under the linear transformation \mathbf{A} .

Similarly, one can extend the notion of eigenvalue and eigenvector for any linear transformation (not necessarily represented by a matrix) to a continuous function. The corresponding equation that defines the eigenvalue problem is similar to Equation (2.13):

$$\mathcal{D}f = \lambda f, \quad (2.22)$$

where the function f is called the *eigenfunction* of the linear operator \mathcal{D} , and the scalar value λ is called the corresponding eigenvalue. The left hand side of the equation represents the linear transformation represented by \mathcal{D} on the function f , while the right hand side of the equation represents the function f scaled by the scalar value λ .

For example, the derivative operator $\frac{d}{dx}$ takes the first order derivative of a function. The eigenfunction of the derivative operator is the exponential function $f(x) = e^{\lambda x}$, *i.e.*,

$$\frac{d}{dx}e^{\lambda x} = \lambda e^{\lambda x}. \quad (2.23)$$

A continuous function can be seen as a vector of infinite size.

We will use calligraphic letters to represent linear operators in the rest of the dissertation.

As one can see, the derivative operator has infinitely many eigenvalues λ . This is different to the case of a matrix whose number of eigenvalues is limited by the dimension of the matrix.

2.1.4 Laplace transform

The Laplace transform¹ of a function $f(t)$, denoted as $\mathcal{L}\{f(t)\}$, is an integral transform that converts the function f from its original domain $t \in \mathbb{R}$, to a function of another variable σ . The transformed function is often denoted as $\hat{f}(\sigma)$.

¹ Named after its inventor, Pierre-Simon Laplace

$$\hat{f}(\sigma) = \mathcal{L}\{f(t)\} = \int_0^{+\infty} f(t)e^{-\sigma t} dt. \quad (2.24)$$

The original domain $t \in \mathbb{R}$ of the function f often represents the time in real applications, while the transformed domain σ is called the σ domain, or sometimes the (complex) frequency domain. In this dissertation we will only deal with situations where σ is also a real variable, i.e., $\sigma \in \mathbb{R}$.

Note f and \hat{f} are two functions defined in two different domains.

2.1.4.1 Linearity

The Laplace transform \mathcal{L} is also a linear transformation.

Let f and g be two continuous functions defined in the same domain $t \in \mathbb{R}$. Let c be any constant scalar. By using the definition of the Laplace transform we can get

$$\mathcal{L}\{f(t)\} + \mathcal{L}\{g(t)\} = \mathcal{L}\{f(t) + g(t)\}, \quad (2.25)$$

$$c\mathcal{L}\{f(t)\} = \mathcal{L}\{cf(t)\}. \quad (2.26)$$

Linearity of the Laplace transform

Similar to the matrix multiplication case in Section 2.1.2, the two equations above show that the Laplace transform fulfils the two properties of linearity: additivity and homogeneity of degree one.

- Equation (2.25) shows that the Laplace transform is additive: The order of the addition and the Laplace transform does not affect the end result.

- Equation (2.26) shows that the Laplace transform is homogeneous of degree one: The order of the constant scaling and the Laplace transform does not affect the end result.

Thus we say that the Laplace transform is a linear transformation.

2.1.4.2 Region of convergence

Now we look at the definition of the Laplace transform again.

$$\hat{f}(\sigma) = \mathcal{L}\{f(t)\} = \int_0^{+\infty} f(t)e^{-\sigma t} dt. \quad (2.24)$$

The Laplace transform of a function $f(t)$ is the integral of the product of the function itself and an exponential function, $e^{-\sigma t}$, over the entire positive real axis $t \in (0, +\infty)$. However, there is no guarantee that this integral converges.

For example, consider the simple constant function

$$f_1(t) = 1. \quad (2.27)$$

The Laplace transform $\hat{f}_1(\sigma)$ is simply the integral of the exponential function $e^{-\sigma t}$:

$$\hat{f}_1(\sigma) = \mathcal{L}\{f_1(t)\} = \int_0^{+\infty} e^{-\sigma t} dt. \quad (2.28)$$

Thus σ must be larger than 0 in order for the integral to converge.

The set of σ for which $\hat{f}(\sigma)$ converges is defined as the *region of convergence* of the Laplace transform. When σ is positive, the term $e^{-\sigma t}$ is a decaying function of t , and the larger σ is, the faster $e^{-\sigma t}$ decays. Thus, intuitively, the larger σ is, the “easier” it is for the Laplace transform $\hat{f}(\sigma)$ to converge. Indeed, if the Laplace transform converges at σ_0 , then it also converges for all $\sigma > \sigma_0$.

For the most general case when complex σ is allowed, the real part is compared, i.e., $\text{Re}(\sigma) > \text{Re}(\sigma_0)$.

Obviously, the region of convergence depends on the growth behaviour of the function $f(t)$. Intuitively, if $f(t)$ grows faster, σ needs to be larger to “compensate” the growth of $f(t)$. However, this is not always true. Consider the function

$$f_2(t) = t^{-1}, \quad (2.29)$$

which, decays faster than the constant function $f_1(t) = 1$. If we take $\sigma = 0$, given the fact that

$$\hat{f}_2(0) = \int_0^{+\infty} t^{-1} dt \quad (2.30)$$

still diverges, the region of convergence for $f_2(t)$ is $(0, \infty)$, still the same as for $f_1(t) = 1$.

The region of convergence can also be empty in some cases. This happens for some $f(t)$ which grow faster than exponentially, for instance, consider

$$y = f_3(t) = \frac{1}{c-t}, \quad t \in (0, c). \quad (2.31)$$

$f_3(t)$ already reaches infinity when t takes the finite limit of $t \rightarrow c^-$. Thus, no σ is large enough to compensate the growth of $f_3(t)$, and its region of convergence is empty.

Consider $f_3(t)$ as the solution to $dy/dt = y^2$, as compared with $dy/dt = \sigma y$ for the exponential function.

2.1.5 Linear time-invariant system

In this subsection, we introduce the linear time-invariant (LTI) system.

2.1.5.1 Convolution

The *convolution* of two continuous functions $f(t)$ and $g(t)$, denoted as $f \star g$, is defined as integral of the product of the two functions after one is reversed and shifted:

$$(f \star g)(t) = \int_{-\infty}^{+\infty} f(\tau)g(t-\tau) d\tau. \quad (2.32)$$

Note the result is still a function of t .

The convolution process is commutative. This can be seen from the fact that it does not matter which one of the functions $f(t)$ and $g(t)$ is reversed and shifted before the integration:

$$(f \star g)(t) = \int_{-\infty}^{+\infty} f(\tau)g(t-\tau) d\tau \quad (2.33)$$

$$= \int_{-\infty}^{+\infty} f(t-\tau)g(\tau) d\tau = (g \star f)(t). \quad (2.34)$$

Commutativity of convolution

2.1.5.2 Linearity and time-invariance

An LTI system is a system that produces an output function $y(t)$ from any input function $x(t)$, subject to the constraints of linearity and time-invariance.

We have explained the linearity of matrix multiplication in Section 2.1.2, and here it is very similar.

- *Linearity* means that the relationship between the input $x(t)$ and the output $y(t)$ is linear, such that the LTI system has the properties of additivity and homogeneity of degree one.

Additivity: If, in an LTI system, the input functions $x_1(t)$ and $x_2(t)$ correspond to the output functions $y_1(t)$ and $y_2(t)$ respectively, then for the input function $x_3(t) = x_1(t) + x_2(t)$, the output function $y_3(t)$ will be the sum of $y_1(t)$ and $y_2(t)$, i.e., $y_3(t) = y_1(t) + y_2(t)$.

Homogeneity of degree one: If the output due to input $x(t)$ is $y(t)$, then the output due to input $c \cdot x(t)$ will be $c \cdot y(t - \Delta t)$ for any given constant scalar c .

Whether we apply an input to the system now or Δt from now, the output will be identical except for a time delay of Δt .

- *Time-invariance* means that the LTI system is invariant under time shift: If the output due to input $x(t)$ is $y(t)$, then the output due to input $x(t - \Delta t)$ will be $y(t - \Delta t)$ for any given time shift Δt .

The *impulse response* of an LTI system, denoted as $h(t)$, is an intrinsic property of this particular system. The impulse response $h(t)$ represents the system's response in the output $y(t)$ corresponding to a unit impulse $x(t) = \delta(0)$ in the input.

δ is the Dirac delta function. $\delta(0)$ can be intuitively understood as "a unit of input" at time $t = 0$.

The output function $y(t)$ represents the total response of the system. Using the linearity and the time-invariance of the LTI system, one can find that the output at given time t equals the integral of the product of $h(t_a)$ and $x(t_b)$ where $t_a + t_b = t$, which is just the convolution of the impulse response $h(t)$ and the input function $x(t)$:

$$y(t) = (h \star x)(t) = \int_{-\infty}^{+\infty} h(\tau)x(t - \tau) d\tau. \quad (2.35)$$

2.1.5.3 Eigenfunctions and eigenvalues of LTI systems

An LTI system with impulse response $h(t)$ can be understood as a transformation from the input $x(t)$ to the output $y(t)$, and this

transformation is linear by definition. Thus an LTI system can also be understood as a linear operator, which we denote as \mathcal{H} :

$$y(t) = \mathcal{H}x(t) = (h \star x)(t) = \int_{-\infty}^{+\infty} h(\tau)x(t - \tau) d\tau. \quad (2.36)$$

The eigenfunction and eigenvalue of an LTI system can thus be defined via

$$\mathcal{H}f = \lambda f, \quad (2.37)$$

where f is the eigenfunction, and the constant λ is the eigenvalue. This means that the output of the LTI system is a scaled version of input function, when the input function is f .

Now we proof that the exponential functions $f = Ae^{st}$ are the eigenfunctions of an LTI operator \mathcal{H} , where A and s are both constants. Starting from Equation (2.36) we can get

$$\mathcal{H}f = \int_{-\infty}^{+\infty} h(\tau)f(t - \tau) d\tau \quad (2.38)$$

$$= \int_{-\infty}^{+\infty} h(\tau)Ae^{s(t-\tau)} d\tau \quad (2.39)$$

$$= Ae^{st} \int_{-\infty}^{+\infty} h(\tau)e^{-s\tau} d\tau \quad (2.40)$$

$$= Ae^{st}\hat{h}(s) = \hat{h}(s) \cdot f, \quad (2.41)$$

where $\hat{h}(s)$ is the Laplace transform of the impulse response $h(t)$, which depends on the constant parameter s in f .

Till now, we have recovered the eigensystem expressed in Equation (2.37), and proved that the exponential functions $f = Ae^{st}$ are the eigenfunctions of an LTI operator \mathcal{H} . We have also seen that the eigenvalues λ of an LTI operator \mathcal{H} are the Laplace transform of its impulse response $h(t)$.

See Section 2.1.3.2 for details about operators and eigenfunctions.

As usual we only deal with real-valued A and s in this dissertation, but in general they can be complex numbers.

See Section 2.1.4 for details about the Laplace transform.

2.2 GRAPHS AND NETWORKS

Entities in the real world are often connected, and graphs or networks are used to model their connections. While multiple definitions of graphs and networks exist [13, 28, 54], there is no standard due to the interdisciplinary nature of the field. In this dissertation we use the similar definitions as defined in [28].

A graph or a network is a pair $G = (V, E)$, where $V = \{v_1, \dots, v_n\}$ is a set of *vertices* or *nodes*, and $E \subseteq V \times V$ is a set of *edges* or *links*. An

edge $e = (v_1, v_2)$ connects two nodes v_1 and v_2 . Strictly speaking, a network is an actual object, and a graph is the underlying mathematical representation of a network. However this difference is subtle and causes no ambiguity in this dissertation. Therefore we will use these terminologies interchangeably.

2.2.1 Network types

Networks can belong to different types depending on the actual use cases. Here we list commonly used network types and their typical use cases.

Directed and undirected networks

For a network $G = (V, E)$, if E contains directed edges, the network is called a *directed network*, or *digraph*. Conversely, if E contains only undirected edges, the network is called an *undirected network*. Note it makes sense to write an undirected edge e as the form $\{v_i, v_j\}$, since the order of v_i and v_j does not distinguish two different edges.

Directed networks are suitable for modelling situations where the direction of a relationship between entities is important. For example, in the Twitter¹ “mentioning” network [20], each node represents a Twitter user, and each directed edge from user A to user B means that user A has mentioned user B in a tweet using the “@username” syntax.

Undirected networks are suitable for modelling situations where a relationship between entities is bidirectional, or the direction of a relationship is of no interest. For example, in the Facebook² friendship network [74], the friendship between two users is bidirectional.

Weighted and unweighted networks

For a network $G = (V, E)$, if the edges in E have weights, the network is called a *weighted network*. Conversely, if the edges in E do not have weights, the network is called an *unweighted network*.

¹ <https://twitter.com>

² <https://www.facebook.com>

Weights can carry different meanings. For instance, in a social network, the weight of an edge can mean the strength of a relationship, or the length of an interaction. In a transportation network, the weight of an edge can mean the volume or the traffic flow of a connection.

Signed and unsigned networks

For a network $G = (V, E)$, if the edges in E have signs (either positive, denoted by “+”; or negative, denoted by “-”), the network is called a *signed network*. Conversely, if the edges in E do not have signs, the network is called an *unsigned network*.

Signed networks can be used to model social situations where friendships and enmities coexist. For example, the Slashdot Zoo network [41] contains directed, signed edges that connect users in the technology news site, Slashdot³. A positive edge from user A to user B means that user A has marked user B as a “friend”, increasing the scores of user B’s posts as shown to user A; while a negative edge from user A to user B means that user A has marked user B as a “foe”, decreasing the scores of user B’s posts as shown to user A.

Multigraphs

For a network $G = (V, E)$, if two nodes can be connected by any number of edges multiple times, the network is called a multigraph [63].

For instance, the Reality Mining network [23] is an undirected multigraph that contains human contact data among 100 students of the Massachusetts Institute of Technology (MIT), collected in a social experiment in 2004 as part of the Reality Commons project. A node represents a person, and an edge represents a physical contact. Multiple edges can exist between a pair of nodes.

2.2.2 *Common terminology*

In this subsection, we introduce the common terminology of graphs and networks that will be used through the dissertation.

³ <https://slashdot.org>

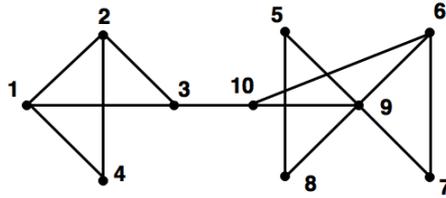


Figure 2.1: The SimGraph network.

Size and volume

The *size* of a network, often denoted by N , is the number of nodes in the network, *i.e.*, $N = |V|$. The *volume* of the network, often denoted by M , is the number of edges in the network, *i.e.*, $M = |E|$.

Simple graphs and simple digraphs

A simple graph is an undirected, unweighted, unsigned graph without loops or multiple edges. Figure 2.1 shows the SimGraph network, an example of a simple graph with $N = 10$, $M = 14$.

Similarly, a simple digraph is a directed, unweighted, unsigned graph without loops or multiple edges.

Terminology for undirected networks

Let $G = (V, E)$ be an undirected network. Two nodes v_1 and v_2 are called *adjacent* to each other, if $\{v_1, v_2\} \in E$, *i.e.*, they are connected by an edge $\{v_1, v_2\}$. The two nodes are then called *neighbours*. The edge $\{v_1, v_2\}$ is called *incident* to both v_1 and v_2 .

The *degree* of a node v , denoted by k , is the number of edges that are incident to v . If G is a simple graph, the degree of a node is equal to the number of its neighbours.

Terminology for directed networks

Let $G = (V, E)$ be a directed network. If $(v_1, v_2) \in E$, *i.e.*, there exists an edge from node v_1 to node v_2 , the node v_1 is called a *parent* of v_2 ,

and v_2 is called a *child* of v_1 . The edge e is called an *outgoing edge* of v_1 , and an *incoming edge* of v_2 .

The *indegree* of a node v , denoted by k^{in} , is the number of incoming edges of v . For a simple digraph, the indegree of a node is equal to the number of its parents.

The *outdegree* of a node v , denoted by k^{out} , is the number of outgoing edges of v . For a simple digraph, the outdegree of a node is equal to the number of its children.

2.3 MATRIX REPRESENTATIONS OF NETWORKS

2.3.1 Adjacency matrix

The *adjacency matrix* \mathbf{A} of a simple graph $G = (V, E)$ with $V = \{v_1, \dots, v_n\}$ is an $N \times N$ matrix defined as follows:

$$\mathbf{A}_{i,j} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \in E \\ 0 & \text{otherwise} \end{cases} \quad (2.42)$$

More generally, the element $\mathbf{A}_{i,j}$ can be the weight of the edge $\{v_1, v_2\}$ for weighted networks, the sign of the edge $\{v_1, v_2\}$ for signed networks, or the number of edges between v_i and v_j for multigraphs. Equation (2.43) shows the adjacency matrix of the SimGraph network.

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix} \quad (2.43)$$

In general, for an undirected network G , the adjacency matrix \mathbf{A} is always symmetric, *i.e.*, $\mathbf{A} = \mathbf{A}^T$.

Similarly, the adjacency matrix \mathbf{A} of a directed network $G = (V, E)$ with $V = \{v_1, \dots, v_n\}$ is an $N \times N$ matrix defined as follows:

$$\mathbf{A}_{i,j} = \begin{cases} 1 & \text{if } (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases} \quad (2.44)$$

More generally, the element $\mathbf{A}_{i,j}$ can be the weight of the edge (v_1, v_2) for weighted directed networks, the sign of the edge (v_1, v_2) for signed directed networks, or the number of edges from v_i to v_j for directed multigraphs.

2.3.2 Degree matrix

The *degree matrix* \mathbf{D} of an undirected network $G = (V, E)$ with $V = \{v_1, \dots, v_n\}$ is an $N \times N$ diagonal matrix defined as follows:

$$\mathbf{D}_{i,j} = \begin{cases} k_i & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \quad (2.45)$$

where k_i is the degree of v_i . Equation (2.46) shows the degree matrix of the SimGraph network.

$$\mathbf{D} = \begin{pmatrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \end{pmatrix} \quad (2.46)$$

Similarly, one can define the *indegree matrix* \mathbf{D}^{in} and the *outdegree matrix* \mathbf{D}^{out} of a directed network $G = (V, E)$ with $V = \{v_1, \dots, v_n\}$. The indegree matrix and outdegree matrix of G are both $N \times N$ matrices defined as follows:

$$\mathbf{D}_{i,j}^{in} = \begin{cases} k_i^{in} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \quad (2.47)$$

$$\mathbf{D}_{ij}^{out} = \begin{cases} k_i^{out} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \quad (2.48)$$

where k_i^{in} and k_i^{out} are the indegree and outdegree of node v_i , respectively.

2.4 DEGREE DISTRIBUTION

The degree distribution of a network is the probability distribution of the node degrees in the entire network.

The degree distribution can be expressed as a discrete function $P(k)$ of the degree k , where $P(k)$ is defined as the fraction of nodes in the network with degree k . For example, the degree distribution of the SimGraph network as in Figure 2.1 is

$$P(k) = \begin{cases} 0.4 & \text{if } k = 2 \\ 0.5 & \text{if } k = 3 \\ 0.1 & \text{if } k = 5 \\ 0 & \text{otherwise} \end{cases} \quad (2.49)$$

Moreover, the degree distribution of a network can also be expressed as $P(K \geq k)$, which is the complementary cumulative distribution function (CCDF) of the degree k . $P(K \geq k)$ is the fraction of nodes in the network that have degree not less than k .

By definition, $P(K \leq k)$ is a monotonically decreasing function of k . $P(K \leq k)$ is also called the *cumulative degree distribution* of the network. For example, the cumulative degree distribution of the SimGraph network as in Figure 2.1 is

$$P(K \geq k) = \begin{cases} 1 & \text{if } k \leq 2 \\ 0.6 & \text{if } 2 < k \leq 3 \\ 0.1 & \text{if } 3 < k \leq 5 \\ 0 & \text{if } k > 5 \end{cases} \quad (2.50)$$

Similarly, one can define the indegree distribution and the outdegree of a directed network.

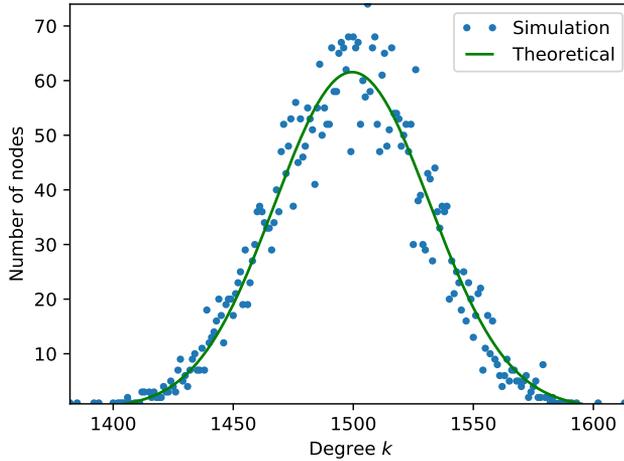


Figure 2.2: The theoretical degree distribution of the Erdős-Rényi model $G(5000, 0.3)$, and the degree distribution of an instance of a synthetic graph generated from the model.

2.4.1 Degree distribution of the Erdős-Rényi model

In the Erdős-Rényi model $G(N, \pi)$, a simple graph is constructed by connecting N nodes with edges randomly. The probability that an edge exists between each pair of nodes is equal to π , independent from every other edge [25].

The Erdős-Rényi model has a binomial theoretical distribution of

$$P(k) = \binom{N-1}{k} \pi^k (1-\pi)^{N-1-k}. \quad (2.51)$$

Figure 2.2 shows the theoretical degree distribution of the Erdős-Rényi model $G(5000, 0.3)$, as well as the degree distribution of an instance of a synthetic graph generated from the model.

2.4.2 Degree distribution of real-world networks

Unlike the Erdős-Rényi model which has a symmetric bell-curved degree distribution, a real-world network usually exhibits a highly

right-skewed, heavy-tailed degree distribution, which means the majority of nodes have low degree, while a small but significant fraction of nodes have significantly high degree.

Empirically, many networks, including the Internet, the World Wide Web, and social networks have degree distributions that approximately follow a power-law [18]:

$$P(k) \sim k^{-\alpha}, \quad (2.52)$$

where α is a constant that typically lies in the range $2 < \alpha < 3$, although studies have shown that the observed degree distributions almost always deviate from perfect power functions [14].

The degree distribution and the cumulative degree distribution of a real-world network are usually visualised using plots with the log-log scale. A pure power-law degree distribution is shown as a straight line in the log-log plot.

Alternatively, one can use “frequency” (*i.e.*, the number of nodes with degree k) instead of “probability” as the Y-axis in the degree distribution plot. Since the frequency is just the probability scaled by the network size N which is constant for a network, using frequency as the Y-axis does not change the shape of the curve in the log-log plot.

As an example, Figure 2.3 shows the degree distribution of Gowalla, a discontinued online social network [17].

2.4.3 Continuous approximation of the degree distribution

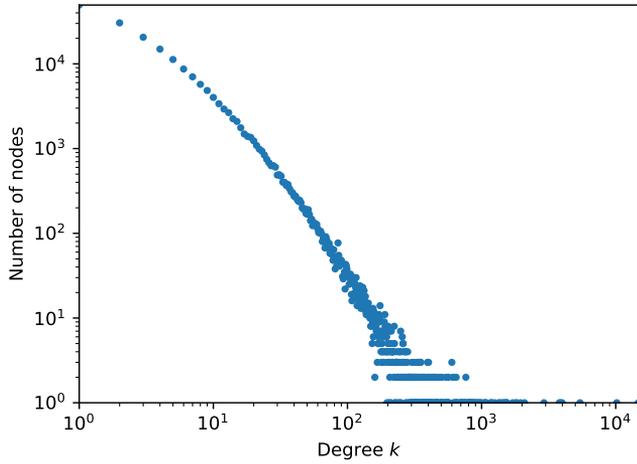
One can use continuous functions to approximate discrete degree distribution functions. For example, the power-law degree distribution as shown in Equation (2.52) can be approximated as

$$p(k) = ck^{-\alpha}, \quad (2.53)$$

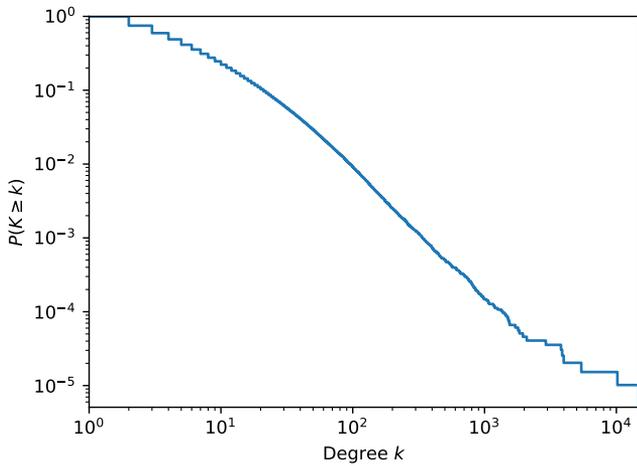
where c is a constant, and the exponent α typically lies in the range (2, 3).

The continuous function $p(k)$ must be a valid probability density function, *i.e.*,

$$\int_{k_{\min}}^{k_{\max}} p(k)dk = 1. \quad (2.54)$$



(a)



(b)

Figure 2.3: The degree distribution of the Gowalla network. Panel (a) plots the number of nodes (frequency) against degree k ; Panel (b) shows the cumulative degree distribution (complementary cumulative distribution function).

If k is assumed to have the range $k \in [1, +\infty)$, one can solve that the constant $c = \alpha - 1$, *i.e.*, the approximated probability density function of the degree distribution function is

$$p(k) = (\alpha - 1) k^{-\alpha}, \quad 2 < \alpha < 3. \quad (2.55)$$

If the cumulative degree distribution $P(K \geq k)$ is also approximated as a continuous function, the approximated probability density function $p(k)$ can also be written as the derivative of $-P(K \geq k)$, *i.e.*,

$$p(k) = -\frac{dP(K \geq k)}{dk}. \quad (2.56)$$

This however requires $P(K \geq k)$ to be differentiable.

2.5 STRUCTURAL PROPERTIES

In this section, we describe various metrics that can characteristic the structural property of a node, and sometimes, of an entire network. Structural properties can be reflected in the structure of the network, for instance the centrality of a node. Other non-structural properties, such as the label of a node, are not reflected directly in the structure of the network.

2.5.1 Degree centrality

The *degree centrality* of a node, denoted as c_i^{deg} , is defined as the degree of the node.

$$c_i^{deg} = k_i. \quad (2.57)$$

The degree centrality reflects the simple idea that a node is more centric if it has more neighbours.

2.5.2 Eigenvector centrality

The idea of node centrality is to differentiate nodes according to how “centric” they are. However, according to the definition of the degree centrality, each neighbour of a node contributes exactly the same amount to the centrality value of the node, regardless of its structural

differences to the others. This somehow contradicts the idea of node centrality.

The eigenvector centrality instead, considers the “quality” of the neighbours: A node is more centric if it has more neighbours, and its neighbours are more centric.

Let $G = (V, E)$ be an undirected network represented by its adjacency matrix \mathbf{A} . The eigenvector of a node, denoted as c_i^{eig} , is proportional to the sum of eigenvector centralities of all its neighbours. This recursive definition can be expressed as follows:

$$c_i^{eig} = \lambda^{-1} \sum_j \mathbf{A}_{i,j} c_j^{eig}, \quad (2.58)$$

where λ is a positive constant, and the eigenvector centralities are all non-negative. To avoid trivial solutions, all eigenvector centralities must not be zero.

If the eigenvector centralities of all nodes column vector \mathbf{v} of size N such that $\mathbf{v}_i = c_i^{eig}$. Equation (2.58) can be rewritten as an eigenvalue problem:

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}, \quad (2.59)$$

where \mathbf{v} is the eigenvector of the adjacency matrix \mathbf{A} , and λ is the corresponding eigenvalue. Considering the above-mentioned other constraints, λ is the dominant eigenvalue of \mathbf{A} .

The eigenvector centrality can be obtained using power iteration, or the eigendecomposition of the adjacency matrix \mathbf{A} .

2.5.3 Random walk centrality

Consider the following *random walk* process of a random walker on a simple digraph $G = (V, E)$. At any timestep t the random walker walks to one of the children of the node v where he is currently at, with equal probability. If v has no child, the random walker randomly teleports to any node in the network with equal probability.

Define the $N \times N$ matrix \mathbf{P} such that

$$\mathbf{P}_{i,j} = \begin{cases} \frac{1}{k_i^{out}} & \text{if } k_i^{out} > 0 \text{ and } (v_i, v_j) \in E \\ 0 & \text{if } k_i^{out} > 0 \text{ and } (v_i, v_j) \notin E \\ \frac{1}{N} & \text{if } k_i^{out} = 0 \end{cases} \quad (2.60)$$

See Section 2.1.3 for details about eigenvectors, eigenvalues and eigendecomposition.

\mathbf{P} is the right stochastic matrix that describes the behaviour of the random walker, with $\mathbf{P}_{i,j}$ denoting the conditional probability that, the random walker will walk to vertex v_i in the next step, given the condition that it is at vertex v_j in the current step. By definition, each row of \mathbf{P} sums up to 1.

Let \mathbf{v} be a row vector of size N that denotes the stationary probability distribution of the random walk process. \mathbf{v} is given by the eigenvalue problem

$$\mathbf{v}\mathbf{P} = \mathbf{v}, \quad (2.61)$$

where \mathbf{v} is the left eigenvector of the right stochastic matrix \mathbf{P} , and the corresponding eigenvalue is exactly 1, which is also the dominant eigenvalue of \mathbf{P} . The value v_i is called the *random walk centrality* of node v_i . The random walk centrality can be seen as a generalisation of the eigenvector centrality, which can be obtained using power iteration.

2.5.4 PageRank

PageRank is a node centrality metric by Page et al. [59], that is initially used in Google to measure the importance of website pages. PageRank is closely related to the random walk centrality.

Define the damping factor $\alpha \in (0, 1)$. At each timestep, the random walker has α probability to the random walk process defined in the previous section. In addition, the random walker has $1 - \alpha$ probability to randomly teleport to any node in the network, with equal probability.

Define the $N \times N$ “Google matrix” \mathbf{G} as

$$\mathbf{G} = \alpha\mathbf{P} + \frac{1 - \alpha}{N}\mathbf{J}, \quad (2.62)$$

where \mathbf{J} is the $N \times N$ all one matrix that guarantees the ergodicity of the random process.

The Google matrix \mathbf{G} is also a right stochastic matrix with each row summing up to 1. \mathbf{G} describes the behaviour of the random walker, with $\mathbf{G}_{i,j}$ denoting the conditional probability that, the random walker will walk or teleport to vertex v_i in the next step, given the condition that it is at vertex v_j in the current step. The stationary probability

distribution \mathbf{p} of the random walk and random teleportation process is given by the eigenvalue problem

$$\mathbf{p}\mathbf{G} = \mathbf{p}, \quad (2.63)$$

where \mathbf{p} is the left eigenvector of the Google matrix \mathbf{G} , and the corresponding eigenvalue is exactly 1, which is also the dominant eigenvalue of \mathbf{G} . The value $pr_i = \mathbf{p}_i$ is called the PageRank of node v_i . PageRank can also be seen as a generalisation of the eigenvector centrality, which can be obtained using power iteration.

2.5.5 Coreness and k -core

An undirected graph G is called a k -degenerate graph if every subgraph of G has a node of degree at most k .

The k -core number κ of an undirected graph G , also known as the *degeneracy* of the graph, is defined as the smallest value of k for which G is k -degenerate [47]. The k -core number of a graph gives the information about how sparse the graph is.

A k -core of an undirected graph G is a maximal connected subgraph of G , in which every node has a degree of at least k . A k -core of a graph with a high k contains the most central nodes in the graph. For instance, the people within a k -core with a high k are important for spreading information in a social network and in building communities [37]. In social networks it would be the largest group of people, which are connected to each other with almost the same degree.

Furthermore, one can define the *coreness* of a node in an undirected graph, denoted as c^{core} . The coreness of a node is defined as the maximum k of the k -core(s) it belongs to [56], *i.e.*, a node v_i has coreness c_i^{core} if v_i belongs to a c_i^{core} -core, but not to any $(c_i^{core} + 1)$ -core of the graph. The coreness of a node can also be seen as a centrality measure.

The k -core number of a network and the coreness of the nodes in the network can be obtained by recursively removing nodes with the smallest degree.

2.5.6 Clustering coefficients

Let $G = (V, E)$ be an undirected network.

The *local clustering coefficient* of a node describes how clustered the neighbours of the nodes are. Let $N_v = \{u \mid \{v, u\} \in E\}$ denote the set of neighbours of a node v . The local clustering coefficient C_v^{local} of the node v is defined as the ratio of the number of edges existing between neighbours over the maximal possible number of edges, *i.e.*,

$$C_v^{local} = \frac{2|\{u, w\} \mid u, w \in N_v, \{u, w\} \in E|}{|N_v|(|N_v| - 1)} \quad (2.64)$$

For a simple graph, $|N_v|$ equals the degree of v .

The *global clustering coefficient* of a network, also called the *transitivity*, describes how clustered the nodes in the entire network are [76]. The global clustering coefficient is defined as the probability that any two incident edges are completed by a third edge to form a triangle, *i.e.*,

$$C^{global} = \frac{|\{u, v, w\} \mid \{u, v\}, \{u, w\}, \{v, w\} \in E|}{|\{u, v, w\} \mid \{u, v\}, \{u, w\} \in E|} \quad (2.65)$$

with $u, v, w \in V$.

The global clustering coefficient can take a value between zero and one, where the value one denotes that all possible triangles are present, and zero denotes a triangle-free network.

Alternatively, one can also use the average local clustering coefficient of all nodes in the network as a metric to describes how clustered the network is.

In this chapter, we describe our work on the topic of phenomena in growing networks.

We start from reviewing related works on preferential attachment network models in Section 3.1, which includes the Barabási-Albert model, the Bianconi-Barabási model, and the relevance decay model. At the same time, we also give remarks on different phenomena that either are explained by these models, or are emergent behaviour of them, such as the rich-get-richer effect and the winner-takes-all effect.

In Section 3.2, we present our empirical findings in real data. Using citation networks as an example here, the degree growth is time-invariant: The average degree of nodes of different age has the same functional dependency on node age regardless of when the nodes have entered the network. This seemingly minor observation is actually not trivial. First of all, preferential attachment models without ageing are known to have a strong first-mover advantage: The first nodes accumulate many more links than the nodes that enter the network later [53]. We show that an accelerated network growth, a feature that is common in real networks [21, 27, 62] yet usually overlooked by network modelling, is an important part of the interplay between preferential attachment and the macroscopic degree growth patterns. In particular, of different growth forms that can be considered, the exponential network growth is consistent with the time-invariant degree growth.

To systematically explore the conditions under which a time-invariant degree growth arises, we introduce a novel mathematical formalism for preferential attachment-based models in Section 3.3, where exponential and linear network growth emerge as the only possible solutions of an eigenvalue problem. Our new formalism also reveals the connection between the Bose-Einstein condensation [9] in the Bianconi-Barabási model [10] and a similar condensation phenomenon seen in the superlinear preferential attachment [40]. Ageing [22, 50] is necessary to recover realistic degree growth curves that are slower

than exponential (e.g., power functions), and can prevent the network condensation from happening.

In Section 3.6, we generate synthetic networks with different parameters to evaluate our analytical results, which include the network growth curve, the node degree growth curve, the estimation of the network degree distribution, and under what circumstances the network undergoes a condensation.

In Section 3.7, we summarise the chapter.

3.1 RELATED WORK

3.1.1 Preferential attachment models

Preferential attachment in networks refers to the mechanism that nodes have the preference to attach to other nodes that are more connected. Preferential attachment can arise from many nature processes. For instance, in social networks [16, 52], well-known people are more likely to make more relations to others. In the context of citation networks [35, 50] and the World Wide Web [1, 5], scientific publications and web pages with higher numbers of citations or incoming links are more likely to be referred to again due to their advantages in publicity.

As a simple example of the preferential attachment network growth models, Albert and Barabási have proposed the well known Barabási-Albert model [3]: The probability that a new node attaches to an existing node v_i is proportional to its degree k_i . One of the realisations of the Barabási-Albert model [12] is as follows.

- The network starts with one node with a self loop at the initial timestep $s = 1$.
- At each later timestep $s > 1$, a new node joins the network and creates a constant number of m undirected links to existing nodes, each with the preferential attachment *selection probability* Π_i proportional to the degree k_i of the existing node i , i.e.,

$$\Pi_i = \frac{k_i}{Z}, \quad (3.1)$$

where Z is the normalisation factor, also known as the *partition function* in some disciplines.

Clearly, in the Barabási-Albert model Z equals the sum of the degrees of all existing nodes in the network. If, for simplicity, we let $m = 1$, since each undirected link corresponds to two degrees, we have

$$Z = \sum_j k_j = 2N - 1. \quad (3.2)$$

Recall that N denotes the size of the network.

Since at each timestep exactly one node joins the network, we can see that $s = N$. The timestep s here can be seen as the *system time*, whose advance is driven by events in the system. We use the function $k_i(s, s_i)$ to denote the degree growth of a node over (system) time, where s_i is the timestep when node v_i joins the network. When $s = s_i$, the initial value of the node degree is 1, *i.e.*,

$$k_i(s_i, s_i) = 1. \quad (3.3)$$

When the network is large enough, we can apply the continuum approximation to obtain the differential equation of the degree growth of a node,

$$\frac{dk_i}{ds} = \Pi_i = \frac{k_i}{Z} \simeq \frac{k_i}{2s}. \quad (3.4)$$

Combining Equations (3.3) and (3.4) we get the node degree growth function in the Barabási-Albert model,

$$k_i(s, s_i) = \left(\frac{s}{s_i} \right)^{1/2}. \quad (3.5)$$

The first-mover advantage

Equation (3.5) is a power-law degree growth function with the exponent $1/2$, in which the degree of a node is determined by, not only the current system time s , but also the system time s_i when the node joins the network. Figure 3.1 illustrates the expected degree growth curves of three nodes that join the network at different times in the Barabási-Albert model. At the same system time s , the older the node (*i.e.*, with a small s_i) is, the higher degree it has. Moreover, even with the same age in system time ($s - s_i$), older nodes still have an advantage over late comers. In other words, the degree growth in the Barabási-Albert model exhibits a strong *first-mover advantage*.

The first-mover advantage is a microscopic phenomenon which happens at the node level. It is the immediate consequence of the positive feedback loop introduced by the generative process of the network

as explained earlier in this subsections, without which the degrees of the nodes that join the network earlier will grow at a much slower rate. Although the first-mover advantage is at the microscopic level, it leads to the emergence of other phenomena at different levels in the network. In the following subsections, we will discuss these emergent phenomena and their relations with the first-mover advantage.

3.1.2 Mean-field estimation of the degree distribution

Barabási et al. [6] have proposed the mean-field method to estimate the degree distribution of the Barabási-Albert model. Assuming s is large enough for us to apply the continuum approximation, the system time when any node joins the network, s_i , is thus uniformly distributed in $(0, s]$, i.e.,

$$P(S < s_i) = \frac{s_i}{s}. \quad (3.6)$$

Recalling Equation (3.5), given a fixed network size s , the degree k_i is an monotonically decreasing function of s_i , thus,

$$s_i = \frac{s}{k_i^2}. \quad (3.7)$$

The cumulative degree distribution is then

$$P(K \leq k_i) = 1 - P(S < s_i) = 1 - k_i^{-2}, \quad (3.8)$$

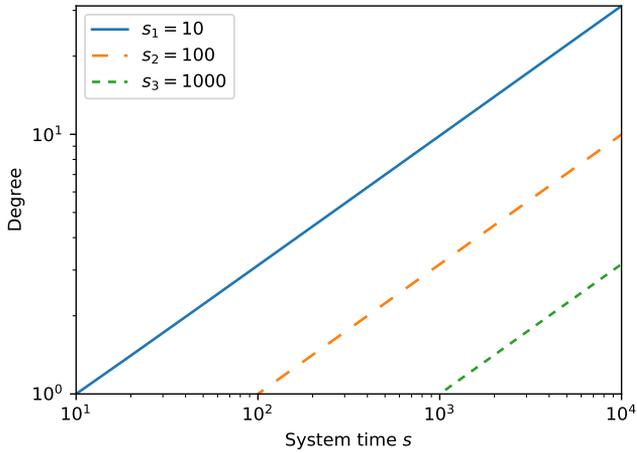
where k_i is in the domain $[1, +\infty)$ can be substituted with k , since it represents the degree of any node. The probability density function of the degree distribution is thus

$$p(k) = \frac{dP(K \leq k)}{dk} = 2k^{-3}, \quad k \in [1, +\infty). \quad (3.9)$$

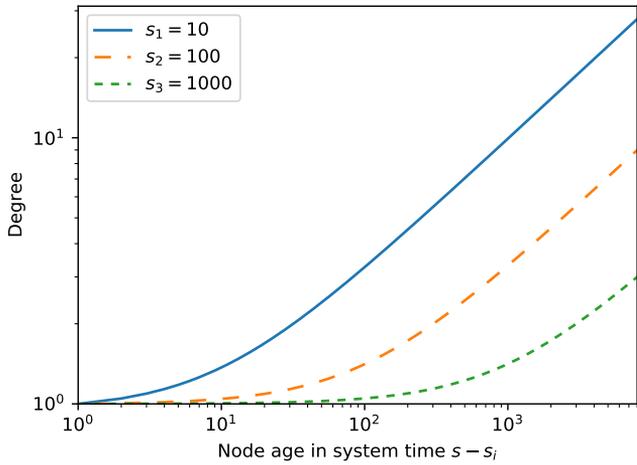
Scale-free degree distribution

Figure 3.2 compares the degree distributions of two synthetic networks with $N = 50000$ with the theoretical degree distributions as estimated with the mean-field method.

In Figure 3.2a the network is generated with the Barabási-Albert model. The approximate straight line shown with the blue dots in the log-log

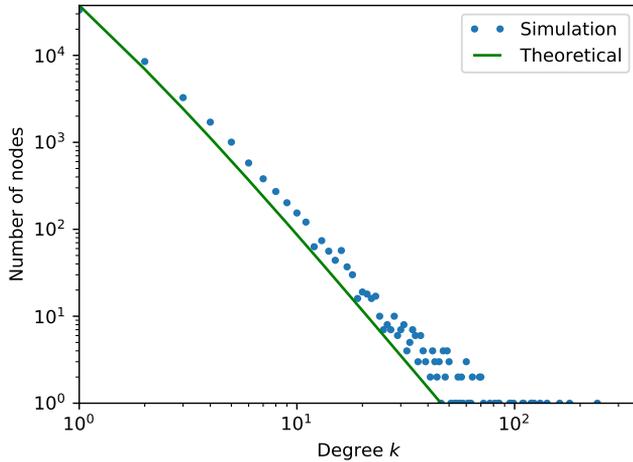


(a)

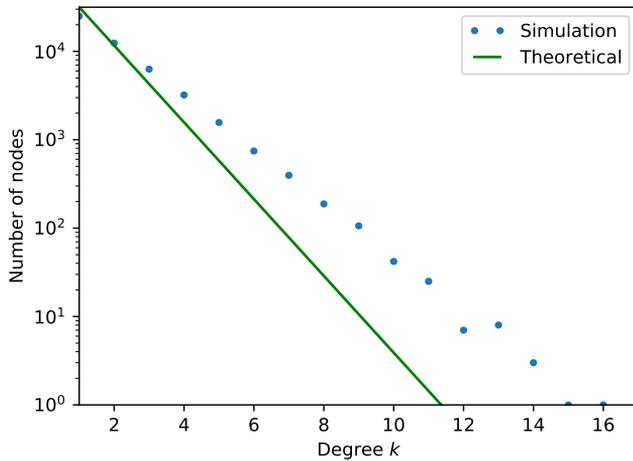


(b)

Figure 3.1: The expected degree growth curves of three nodes that join the network at different times in the Barabási-Albert model, as functions of the (a) system time; (b) node age.



(a)



(b)

Figure 3.2: The degree distributions of two synthetic networks with $N = 50000$, (a) of the Barabási-Albert model; (b) where preferential attachment is absent, and the selection probability Π is the same for all nodes despite of their degrees. Note (a) uses the log-log scale, while (b) uses the linear-log scale.

plot indicates that the node degrees have a power-law distribution, or a *scale-free* distribution [5]. This is in line with the theoretical estimation $p = 2k^{-3}$ as indicated with the green curve.

As a comparison, in Figure 3.2b the network is generated without the presence of the preferential attachment mechanism, *i.e.*, the selection probability Π is the same for all existing nodes despite of their degrees. Figure 3.2b uses the linear-log scale. The blue dots which show as an approximate straight line in the linear-log plot indicate that the degree distribution of the synthetic network is of the exponential form. Note although the mean-field estimation also yields an exponential degree distribution $p = e^{-k+1}$ (shown as the green curve), it shows a significant deviation from the synthetic result. This is because the degree k has a significantly narrower domain as compared with the network generated with the Barabási-Albert model.

The scale-free property of the degree distribution is a macroscopic phenomenon, because the degree distribution is a property of the entire network. The scale-free property is, however, a result of the rich-get-richer effect which is a microscopic phenomenon: As we have seen, the absence of the preferential attachment mechanism results in networks with no scale-free property, which is in line with the theoretical estimation.

This is an important example where phenomena at two levels, microscopic and macroscopic, are directly related. Mathematically, the key variable that connects the two levels is the partition function Z , because it is the sum of all different microscopic “states”², yet itself is a property of the entire system. Thus, the study of the partition function Z plays a central role in preferential attachment models. In the following sections, we will show how to evaluate the partition functions in different generalisations of the Barabási-Albert models.

² The German word of the partition function is “Zustandssumme”.

3.1.3 Bianconi-Barabási model

The Barabási-Albert model successfully explains the scale-free property of networks. It predicts that the degrees of all nodes grow with a power function with a fixed exponent $1/2$, therefore old nodes are expected to remain more popular than late comers. However, in real networks, degrees of nodes grow with different rates, and often we see new nodes get more popular than old ones.

The Bianconi-Barabási model [10] is a natural extension of the Barabási-Albert model: The preferential attachment selection probability Π_i of

an existing node v_i becomes proportional to the product of its degree k_i , and its fitness η_i , *i.e.*,

$$\Pi_i = \frac{k_i \eta_i}{Z}, \quad (3.10)$$

where the normalisation factor Z can be expressed as the sum over all products of degrees and fitness, *i.e.*,

$$Z = \sum_j k_j \eta_j. \quad (3.11)$$

Fitness [15] is an intrinsic property of a node that quantifies its ability to acquire new links in the network. Nodes with higher fitness are more likely to attract links, thus their degrees tend to grow faster. When a node joins the network, it is assigned a fitness value η which does not change over time. Node fitness is typically drawn from some probabilistic distribution $\rho(\eta)$ which is often assumed to be consistent over time. The shape of $\rho(\eta)$ is an important constituent of the network model. At the micro level, $\rho(\eta)$ allows nodes of the same age to grow at different rates. At the macro level, $\rho(\eta)$ affects the broadness of the resulting degree distribution [10]. Studies have been done to determine node fitness values that best correspond to the observed data [49, 75].

Following a derivation similar to Equation (3.5), one can obtain the degree growth function of a node in the Bianconi-Barabási model:

$$k_i(s, s_i, \eta_i) \sim \left(\frac{s}{s_i} \right)^{\beta(\eta_i)}, \quad (3.12)$$

where η_i is the fitness of the node. One can realise that the degree growth function is still a power function, but with the exponent $\beta(\eta_i)$ that is proportional to the fitness of the node, *i.e.*,

$$\beta(\eta) = \frac{\eta}{C}, \quad (3.13)$$

where

$$C = \int d\eta \rho(\eta) \frac{\eta}{1 - \beta(\eta)} \quad (3.14)$$

is a constant for a particular fitness distribution $\rho(\eta)$.

With Equation (3.12), the normalisation factor of the Bianconi-Barabási model can be rewritten as a function of s :

$$Z(s) = \sum_j k_j \eta_j \simeq \int d\eta \rho(\eta) \eta \int_1^s \left(\frac{s}{s_j} \right)^{\beta(\eta)} ds_j. \quad (3.15)$$

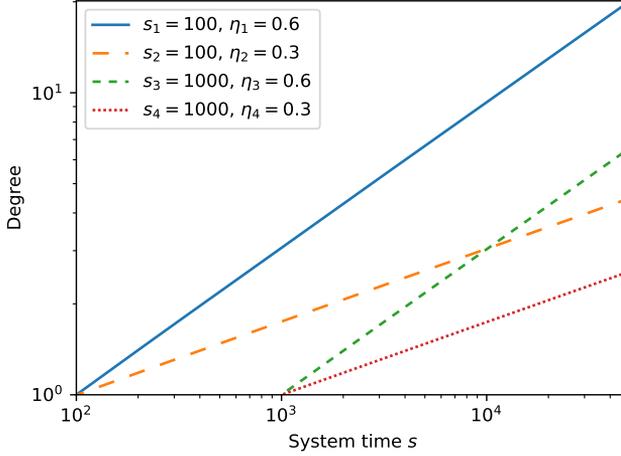


Figure 3.3: The expected degree growth curves of four nodes with different fitness and birth time in the Bianconi-Barabási model. The slopes of the growth curves depend solely on the fitness values, allowing the degrees of nodes with later birth time to surpass earlier ones. However, the strong first-mover-advantage is still present, because the degrees of nodes with the same fitness values still strongly depend on their birth time. Note this plot uses the log-log scale.

And in the limit $s \rightarrow +\infty$, we have

$$\lim_{s \rightarrow +\infty} Z(s) = s \int d\eta \rho(\eta) \frac{\eta}{1 - \beta(\eta)} = sC, \quad (3.16)$$

i.e., the normalisation factor Z is asymptotically proportional to the network size s .

Applying the mean-field estimation as in Section 3.1.2, one can obtain the degree distribution of the Bianconi-Barabási model:

$$p(k) = \int d\eta \rho(\eta) \frac{C}{\eta} k^{-\left(\frac{C}{\eta}+1\right)}. \quad (3.17)$$

As one can realise, the degree distribution $p(k)$ is a weighted sum of multiple power functions, $k^{-(C/\eta+1)}$, and is determined by the fitness distribution $\rho(\eta)$. For most fitness distributions, the resulting degree distribution $p(k)$ still reflects the scale-free property of real world networks, although in general it is not a perfect power function.

3.1.4 Bose-Einstein condensation in networks

The constant C in the Bianconi-Barabási model is the solution to Equation (3.14). However, whether such a solution exists depends on the actual fitness distribution $\rho(\eta)$: When $\rho(\eta)$ is extremely skewed, Equation (3.14) lacks a solution.

To study exactly which kind of fitness distributions can result in the lack of a solution of Equation (3.14), Bianconi et al. [9] have proposed to borrow the ideas from quantum physics which has been used to study the Bose-Einstein condensation phenomenon by physicists.

3.1.4.1 Bose-Einstein statistics

Named after
Satyendra Nath Bose
and Albert Einstein,
two main
contributors of the
theory.

In quantum statistics, Bose-Einstein statistics are used to describe the distribution of a collection of non-interacting and indistinguishable *bosons* in different quantum states at thermodynamic equilibrium [34].

A boson is a particle with integer spin. Examples of bosons are some elementary particles such as photons, and some composite particles such as mesons and helium-4. The other kind of particles in quantum mechanics is *fermions* which have half-integer spin. Examples of fermions are some elementary particles such as electrons, and some composite particles such as neutrons.

Unlike fermions which obey the Pauli exclusion principle, *i.e.*, two or more identical fermions cannot occupy the same quantum state at the same time, there is no restriction on the number of bosons that can occupy the same quantum state for bosons. This is an important property of bosons which allows the Bose-Einstein condensation to happen, and allows us to map a network to a Bose gas later.

In a quantum system, an *energy level* is one or more quantum states that are associated with the same energy value. The number of quantum states with the same energy is called the *degeneracy* of the corresponding energy level. Particles can occupy an energy level with a certain probability. In Bose-Einstein statistics, the expected distribution of bosons in different energy levels can be given as:

$$\bar{n}_i = \frac{g_i}{e^{(\epsilon_i - \mu)/k_B T} - 1}, \quad (3.18)$$

where \bar{n}_i is the number of bosons in energy level i over the total number of bosons in all energy levels, g_i is the degeneracy of energy

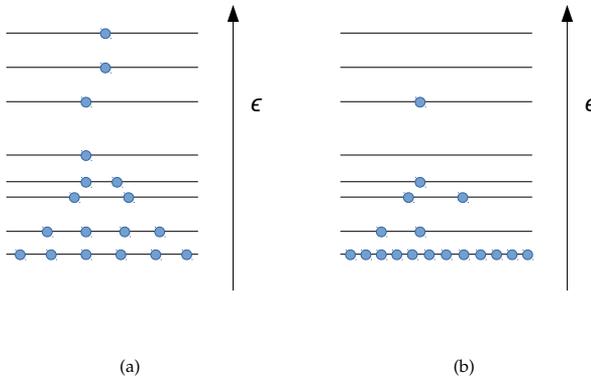


Figure 3.4: Illustration of the energy diagrams corresponding to a Bose gas (a) under high temperature; (b) in the Bose-Einstein condensation phase. Each energy level is associated with an energy ϵ . In (a) the bosons are distributed over a wide range of energy levels. In (b) almost all bosons land in the lowest energy level, while only few bosons populate the higher energy levels sparsely.

level i , μ is the chemical potential of the system, $\epsilon_i > \mu$ is the energy of energy level i , k_B is the Boltzmann constant, and T is the absolute temperature which is non-negative.

Since \bar{n}_i is a monotonically decreasing function of the energy ϵ_i , lower energy means higher chance to be occupied. Moreover, the temperature T can also influence the distribution. High temperature increases the chance of higher energy levels to be occupied, and bosons are distributed in a wider range of energy levels (Figure 3.4a). On the contrary, under low temperature, bosons tend to occupy lower energy levels. Einstein [24] further predicted that when the temperature falls down under a critical temperature $T < T_C$, nearly all bosons “condense” into the lowest energy level (Figure 3.4b). This is called the Bose-Einstein condensation [34]. The actual Bose-Einstein condensate was first experimentally produced in 1976 [66]. Matters in the Bose-Einstein condensation phase can have special properties such as superfluidity and superconductivity [46].

3.1.4.2 Mapping a network to a Bose gas

To use Bose-Einstein statistics to study the Bianconi-Barabási model, it is necessary to map a network of the Bianconi-Barabási model to a Bose

gas, that is, to map each element in the network to its corresponding analogue in a Bose gas.

First, each node with fitness η is associated with an energy level with the energy ϵ such that

$$\epsilon = -T \log \eta, \quad (3.19)$$

where T is the temperature. Nodes with larger fitness have lower energy levels. This is analogous to particles preferring to land on low energy levels. For a certain energy distribution, the temperature T is a parameter to tune the fitness distribution $\rho(\eta)$: When T is lower, the fitness distribution $\rho(\eta)$ is more skewed. The Boltzmann constant k_B is omitted here.

The arrival of a new node v_i to the network corresponds to the addition of a new energy level ϵ_i and two bosons to the system. Of the two bosons which correspond to the two endpoints of the newly created edge, one lands on the energy level of the new node itself, ϵ_i ; while the other one randomly lands on one of the existing energy levels, ϵ_r , with the probability

$$\Pi_r = \frac{e^{-\epsilon_r/T} k_r}{Z(s)} = \frac{e^{-\epsilon_r/T} k_r}{\sum_j e^{-\epsilon_j/T} k_j}, \quad (3.20)$$

which is equivalent to Equation (3.10) in the Bianconi-Barabási model. Deposited bosons are not allowed to jump to other energy levels. In the limit $s \rightarrow +\infty$, the normalisation factor is proportional to s , *i.e.*,

$$\lim_{s \rightarrow +\infty} Z(s) = s \int d\epsilon \varrho(\epsilon) \frac{e^{-\epsilon/T}}{1 - f(\epsilon)} = sC, \quad (3.21)$$

where $\varrho(\epsilon)$ is the energy distribution.

The degree growth function of a node with energy ϵ_i can be solved as

$$k_i(\epsilon_i, s, s_i) = \left(\frac{s}{s_i} \right)^{f(\epsilon_i)} \quad (3.22)$$

with

$$f(\epsilon) = e^{-(\epsilon - \mu)/T}, \quad (3.23)$$

where μ is the chemical potential of the system that satisfies

$$e^{-\mu/T} = \lim_{s \rightarrow +\infty} \frac{Z(s)}{s} = C. \quad (3.24)$$

Combining Equation (3.21) and (3.24) one can get

$$\int d\epsilon \rho(\epsilon) n(\epsilon) = 1 \quad (3.25)$$

with

$$n(\epsilon) = \frac{1}{e^{(\epsilon-\mu)/T} - 1}. \quad (3.26)$$

Equation (3.25) corresponds to an ideal Bose gas of volume 1, where $n(\epsilon)$ is the occupation number of the energy level with energy ϵ , and $n(\epsilon)$ follows Bose statistics in the limit $s \rightarrow +\infty$.

3.1.4.3 Bose-Einstein phase transition

From Equation (3.25) one can realise that, for a certain energy distribution $\rho(\epsilon)$, the chemical potential μ is determined by the temperature T , and their relation can be written as some monotonically decreasing function $T(\mu)$.

However, to guarantee that $n(\epsilon)$ is non-negative for all energy levels, the chemical potential μ cannot be larger than the minimum energy, *i.e.*, $\mu_{\max} = \epsilon_{\min}$. Therefore T has the minimum value of $T(\epsilon_{\min})$ such that Equation (3.25) has a solution for μ . When the temperature decreased lower than $T(\epsilon_{\min})$ however, Equation (3.25) lacks a solution.

The lack of a solution of Equation (3.25) corresponds to the Bose-Einstein condensation in quantum statistics. A Bose gas undergoes the Bose-Einstein phase transition under a critical temperature T_{BE} , and all bosons in the gas land on the lowest energy level as a result. In the Bianconi-Barabási network model, this means there is one node in the network that attracts almost all newly arriving nodes [9].

An example of the energy distribution that results in the Bose-Einstein condensation is the power-law distribution, *e.g.*,

$$\rho(\epsilon) = \frac{\theta + 1}{\epsilon_{\max}^{\theta+1}} \epsilon^\theta, \quad (3.27)$$

where the exponent θ is a free parameter and the energy ϵ takes values in $(0, \epsilon_{\max})$. The lower bound of the critical temperature T_{BE} can then be found as [9]:

$$T_{\text{BE}} > \epsilon_{\max} [\zeta(\theta + 1) \Gamma(\theta + 2)]^{-1/(\theta+1)}. \quad (3.28)$$

However, if a certain energy distribution $q(\epsilon)$ is given, the temperature T is merely a parameter that tunes the fitness distribution $\rho(\eta)$. Thus, the network can as well undergo a Bose-Einstein phase transition without having the analogue of a Bose gas, for instance for the fitness distribution

$$\rho(\eta) = (\lambda + 1)(1 - \eta)^\lambda \quad (3.29)$$

with $\lambda > \lambda_{\text{BE}} = 1$.

3.1.4.4 Similar condensation

A similar condensation phenomenon is seen in the superlinear preferential attachment model [40] as well:

$$\Pi_i = \frac{k_i^\gamma}{Z(s)} = \frac{k_i^\gamma}{\sum_j k_j^\gamma}, \quad \gamma > 1, \quad (3.30)$$

which is very similar to the Barabási-Albert model with linear preferential attachment, but with an exponent $\gamma > 1$ which enhances the rich-get-richer effect.

The superlinear preferential attachment model does not have an asymptotic degree distribution as in the linear preferential attachment case, because the normalisation factor $Z(s)$ does not converge to a value that is proportional to the network size s , as in the linear preferential attachment in Equation (3.1). As a result, eventually a single node connects to nearly all other nodes [40], becoming the absolute hub in the network. However when the exponent γ is close to 1 and the network size is small, the condensation phenomenon may not be extreme, and the network may still exhibit a near scale-free nature before the network size reaches the thermodynamic limit.

Figure 3.5 shows the degree distribution of an instance of a synthetic graph generated from the superlinear preferential attachment model $\Pi \sim k^{1.2}$ with 5×10^4 nodes. Compared with the green line which indicates the theoretical degree distribution of the Barabási-Albert model with linear preferential attachment, the superlinear preferential attachment model results in a more skewed degree distribution, and an absolute hub node is present, with a degree of the same magnitude as the network size, 10^4 .

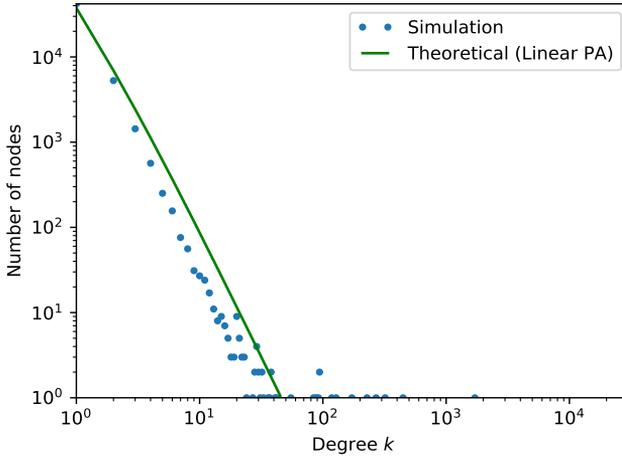


Figure 3.5: The degree distribution of an instance of a synthetic graph generated from the superlinear preferential attachment model $\Pi \sim k^{1.2}$ with 5×10^4 nodes. The green line indicates the curve of the theoretical degree distribution of the Barabási-Albert model with linear preferential attachment.

The winner-takes-all effect

As we have seen in the Bose-Einstein phase transition in the Bianconi-Barabási model and the superlinear preferential attachment model, as the network grows larger, eventually there is a dominant node which occupies a large fraction of all degrees. This is the so called *winner-takes-all* effect, which is an extreme case of the rich-get-richer effect.

In the Bianconi-Barabási model, the rich-get richer effect is amplified by the node fitness. Nodes with higher fitness attract links more efficiently, and the fitness distributions determines whether the winner-takes-all effect occurs. In the superlinear preferential attachment model, the exponent $\gamma > 1$ makes the rich-get richer effect more dominant. Later, we will reveal the deep connection between the Bose-Einstein condensation and the similar condensation effect as in the superlinear preferential attachment model, in order to explain why they both cause the winner-takes-all effect.

3.1.5 *Relevance decay model*

When considering the attractiveness of a node in reality, one typically does not only consider its current degree and its intrinsic quality. Often there is an additional ageing factor that decays over time. For example, when a scientific paper is published, its citation number, which can be modelled as its indegree, is at its minimum. However, its novelty might be at its peak and decays as other new papers are published. When a technology is released, the number of adopters, which can be modelled as its indegree, is at its minimum, while the advancement of the technology might be the highest. However, due to the rapid change of technologies, such advancement might quickly lose relevance.

However, this temporal decaying effect is not considered in the Barabási-Albert model and the Bianconi-Barabási model: The degree of a node never decreases, and the fitness of a node is a constant.

Medo et al. have proposed the use of *relevance* to model this decaying effect [50], where the temporal decay of node attractiveness is modelled by a monotonically decreasing function $R(\tau)$, where τ is the age of a node, defined as $\tau_i := s - s_i$. The preferential attachment probability is proportional to the product of the node degree k , fitness η and the ageing factor $R(\tau)$:

$$\Pi_i = \frac{k_i \eta_i R(\tau_i)}{\sum_j k_j \eta_j R(\tau_j)}. \quad (3.31)$$

The authors used the notion “relevance” as a single term in his original work, but later they preferred to split relevance into fixed fitness and an ageing term. We stick with the latter choice: The product of fitness and the ageing factor is referred to as “relevance”.

As in the Bianconi-Barabási model, the fitness distribution $\rho(\eta)$ is consistent over time. The ageing factor $R(\tau)$ is a monotonically decreasing function of the node age τ that has the initial value $R(0) = 1$, and is the same for all nodes.

In the original work of Medo et al., to ensure the self consistency of the model, the ageing function $R(\tau)$ must decay sufficiently fast (*i.e.*, faster than τ^{-1}), and diminishes to 0 at $\tau \rightarrow +\infty$. A typical ageing function is the exponential decay function $R(\tau) = \exp(-\beta\tau)$.

The above-mentioned requirements of the decay function make the partition function of the model $Z = \sum_j k_j \eta_j R(\tau_j)$ stabilise to a constant

Ω^* when the network is large enough. Moreover, the node degree k must grow sublinearly with the network size, and the actual expected degree growth function is determined by Ω^* and the relevance decay function R :

$$k_i = \exp\left(\frac{\eta_i}{\Omega^*} \int_0^{+\infty} R(\tau) d\tau\right). \quad (3.32)$$

Medo et al. have reported that the relevance decay model can result in realistic degree distributions, such as exponential, log-normal and power-law distributions, depending on the input parameters [50]. They have validated with maximum likelihood methods that the relevance decay model proposed in [50] is the preferential attachment model that best explains the linking patterns in real systems [49].

The ageing effect

The ageing effect reflects the natural preference for newness. Ageing limits the growth of old nodes, thus limiting the rich-get-richer effect, causing the degree growth of a node to slow down as compared with the situation where ageing is absent. In most real systems, it is interesting to study how growth and ageing coexist and interact. For instance in an online platform, if ageing is weak, new users will be discouraged from contributing. On the contrary, if ageing becomes dominant, all contributions get quickly forgotten.

Ageing is a microscopic phenomenon that happens at the node level, although it can also affect the macroscopic behaviour of the system, for example the degree distribution. Later we will show that ageing can prevent the winner-takes-all effect under mild conditions in our model settings.

3.2 EMPIRICAL OBSERVATIONS

To begin our study, we perform empirical studies of the growth patterns in real data in this section.

We first briefly introduce the two scientific citation datasets that we use in this study. We then explore the data by making empirical observations. There are two focuses that we bare in mind when exploring the data. At the micro level, we observe the indegree growth patterns in each dataset, *i.e.*, the growth patterns of citation numbers that the scientific papers receive. We also observe how the average

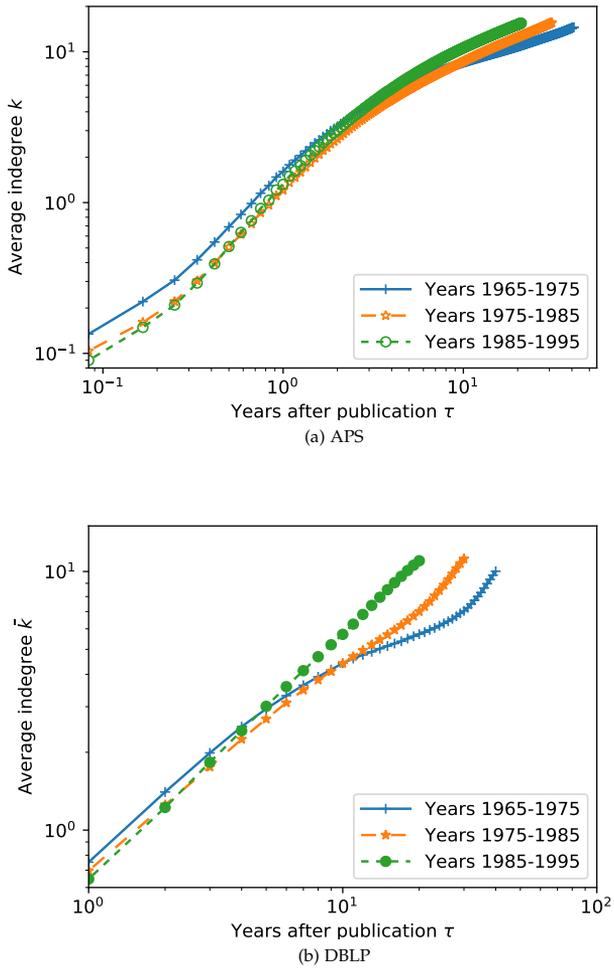


Figure 3.6: The average number of citations as a function of paper age for papers published in different time periods in (a) the APS data and (b) the DBLP data. Papers are grouped by their publication year. Note that the plots use the log-log scale.

outdegree has changed over time in each dataset, *i.e.*, the change of the average citation number that each scientific paper makes. At the macro level, we observe the growth patterns of the network sizes, as well as the degree distributions of the networks.

3.2.1 Datasets

Two citation networks are used in the empirical study, in particular, the American Physical Society (APS) citation network and the DBLP citation network.

In the network representation, a citation between two papers corresponds to a directed link between two nodes in the network. The node outdegree is determined at the moment when the paper together with its list of references is published. By contrast, the node indegree gradually grows from the initial zero value. In terms of growth, we thus focus on the node indegree here.

APS citation network

The APS citation network¹, comprises 564,517 papers that were published from 1893 to 2015 in the APS journals, including Physical Review Letters, Physical Review, and Reviews of Modern Physics. There exist 6,715,562 citations among the papers. The paper publication dates are available with the time resolution of one day.

DBLP citation network

The DBLP Computer Science Bibliography [44] website indexes scientific papers that are published in computer science journals and conferences. In this study, we use the citation network² extracted by Tang et al. in the ArnetMiner project [73].

The DBLP citation network comprises 3,272,991 computer science papers published from 1936 to 2016 and 8,466,859 citations among them. The paper publication dates are available with the time resolution of one year. Note since the original DBLP index is incomplete, there exist missing links in DBLP. The dataset is thus only a sample of the whole citation network.

¹ Available from <https://journals.aps.org/datasets>.

² Available from <https://aminer.org/citation>.

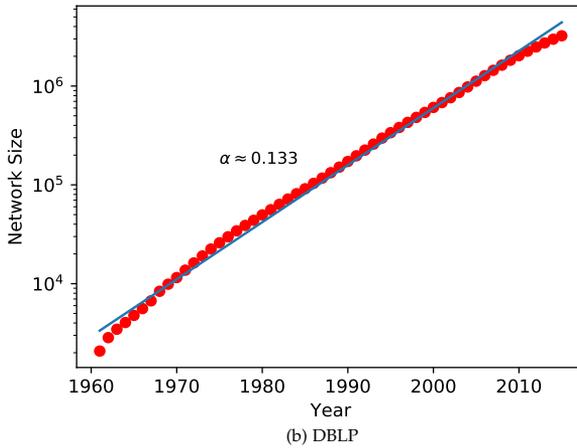
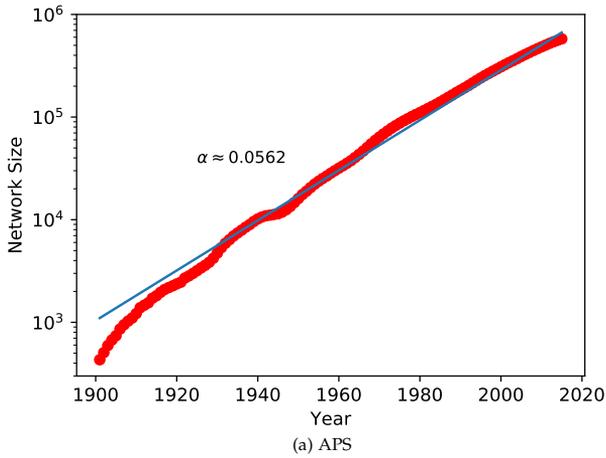


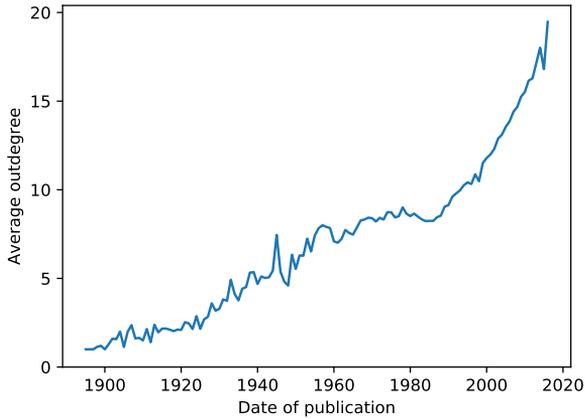
Figure 3.7: The network growth curve of (a) the APS data and (b) the DBLP data. Each plot shows how the total number of papers grows with time in each dataset. Note that the plots use the linear-log scale.

3.2.2 *Time-invariant degree growth*

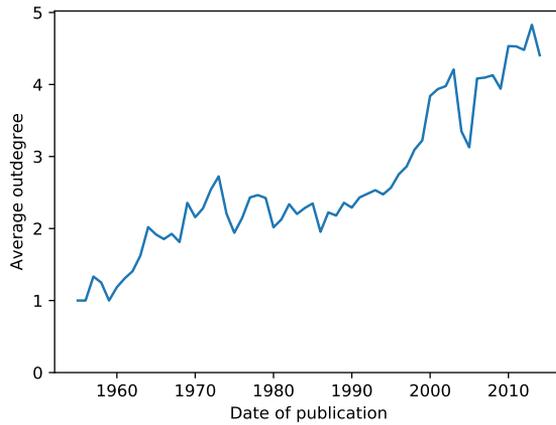
In Figure 3.7, we group the nodes by their publication date and plot the average indegree as a function of the node age separately for nodes originating from different periods. The average paper outdegree is now much higher than it was 50 or more years ago. To limit the impact of this effect, we focus on the time period 1965–1995 during which the average outdegree of papers changed little (see Section 3.2.3). Albeit the individual curves correspond to papers whose publication dates differ by up to 30 years, their shape is strikingly similar. For the APS, we see various curves collapsing onto each other. This indicates that the manner in which the papers’ average number of citations grow with paper age is time-invariant. While the curves’ shapes are more complex for the DBLP data, they are still time-invariant for paper age less than approximately 20 years. In particular, old nodes do not have an advantage over the new ones, compared with Figure 3.11 in which the early-mover advantage is forceful and, in turn, the growth of node degree is strongly determined by the time in which a node appears. These results show that the indegree growth function k is time-invariant: It can be written as a function of the node age τ regardless of the node’s appearance time.

3.2.3 *Varying average outdegree over time*

The growth of the average outdegree of papers with time (see Figure 3.8 for the results in the two studied datasets) can be included in the model but it would come at the cost of increasing the model complexity. Instead, we first limit the impact of the varying average outdegree on the empirical observations presented in Figure 3.7 by focusing on the period 1965–1995 during which the average outdegree changes relatively little. One possible way to further limit such impact is to measure the indegree growth using rescaled indegree which divides the number of new citations in year y by the average outdegree in this year and sums the contributions from individual years. Figure 3.9 shows that rescaled indegree yields similar time-invariant growth patterns as measured using simple indegree. In particular, the average rescaled indegree \tilde{k} in 10 years after publication are 1.04, 1.06 and 1.11 (APS), and 1.99, 1.97 and 2.04 (DBLP), respectively, for the three time periods shown in the figure. In comparison with Figure 3.7, the growth curves have power-law shapes over a broader range of paper age τ .



(a) APS



(b) DBLP

Figure 3.8: The average outdegree of papers published in different years in (A) the APS data and (B) the DBLP data.

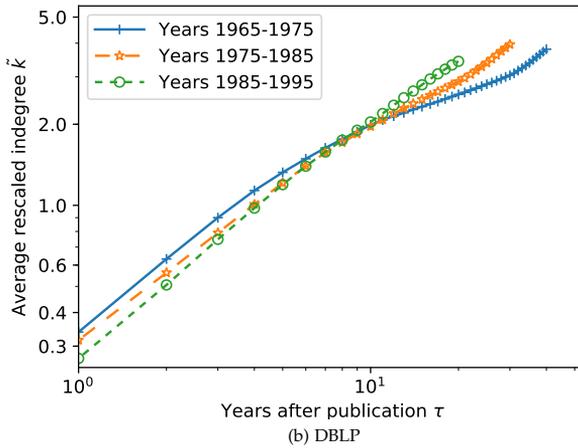
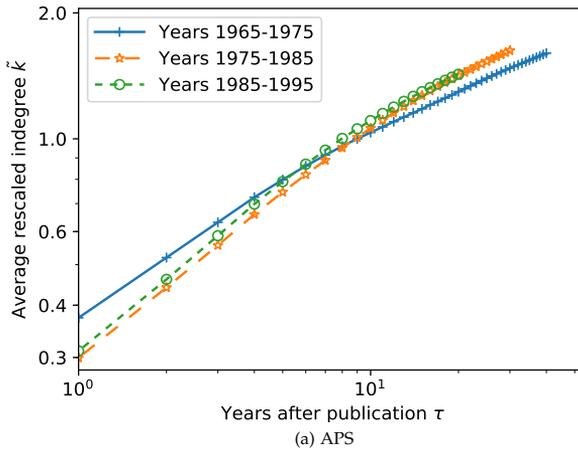


Figure 3.9: The average number of citations as a function of the paper age, rescaled by the average outdegree in the years when the citations were received. Papers are divided in three groups by their publication year.

3.2.4 Exponential network growth

Figure 3.7 shows the evolution of the overall network sizes (measured by the number of nodes) of the two datasets. The near straight lines in the linear-log plots both point at an approximately exponential growth of the network sizes s with time t , *i.e.*,

$$s = e^{at}. \quad (3.33)$$

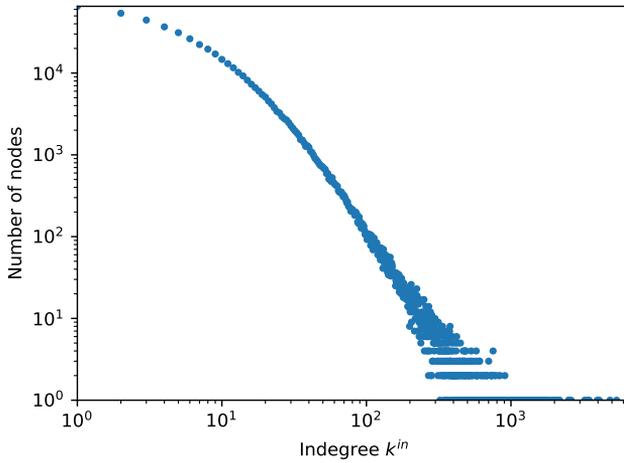
In the original forms of the preferential attachment models that we reviewed in Section 3.1, all times of events are measured in terms of the network size s : from the derivative of the degree growth dk/ds , to the age of the node $s - s_i$. This is, however, not the same as the actual “time” as measured in minutes, hours, days, years, etc., that we observe daily in the physical world, which we define as the *physical time*, denoted t . The network size s , on the other hand, can be seen as the *system time*, which is driven by events in the system, for instance the arrival of new nodes.

The exact relationship between s and t is however not considered in the original preferential attachment models. Thus, there is a natural gap between the physical time, which we directly observe in the real world, and the system time, which the models use to try to “explain the reality”. If the mapping $s = f(t)$ is linear, the models can reflect the reality. Unfortunately as shown in Figure 3.7, this mapping is rather non-linear in the real data. Thus, in this dissertation, as a fundamental step towards a more precise network model, we fill the gap and make a clear distinction between the two kinds of times.

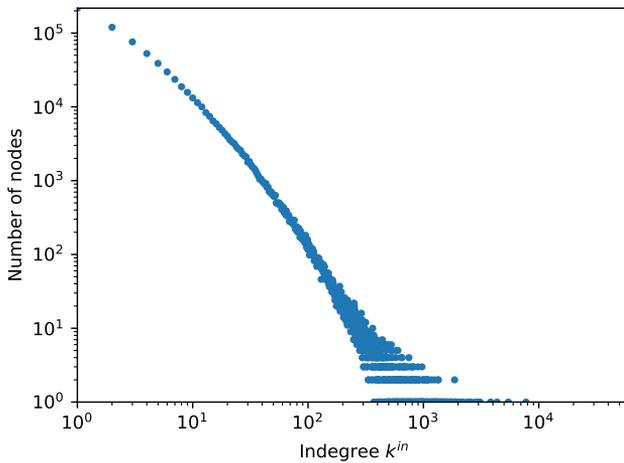
3.2.5 Indegree distribution

Figure 3.10 shows the indegree distribution of the two networks. Several observations can be made from the plots. The majority of papers have low citation numbers, while a small but non-negligible fraction of papers have significantly high citation numbers. The two distributions display power-law like behaviour, although not in the range of small degrees in APS. Additionally, both curves have well-defined long tails.

We use the method by Clauset et al. [18] to estimate the power-law exponents of the two distributions and have found that they have comparable values (1.906 for APS and 1.947 for DBLP).



(a) APS



(b) DBLP

Figure 3.10: The indegree distributions of (a) the APS data and (b) the DBLP data. Estimated power-law exponents (for $k > 5$) are: 1.906 for APS and 1.947 for DBLP.

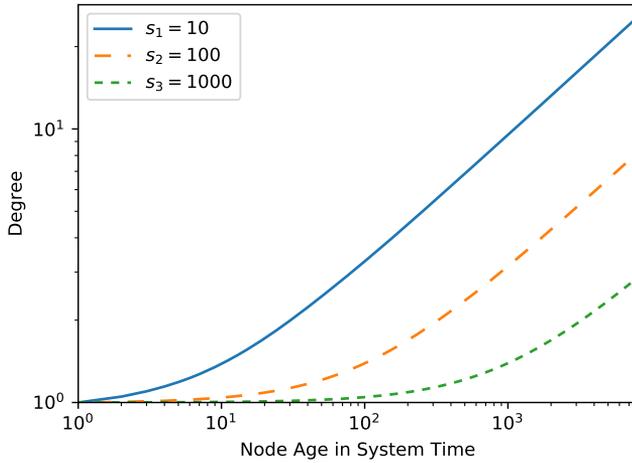


Figure 3.11: The expected degree growth curves of three nodes which join the network at different times (with the same fitness value) in the Bianconi-Barabási model.

3.3 GENERAL PREFERENTIAL ATTACHMENT NETWORK MODEL

In Section 3.1 we reviewed different existing preferential attachment network models that try to explain various phenomena in the real world. While they each successfully explains *some* phenomena, in this section we argue that none of them can explain all observations in Section 3.2.

We now propose a general preferential attachment network model that tries to explain all observations in Section 3.2. Note while the described real datasets are represented with directed networks, we focus here on undirected network models which attract more general interest than directed ones. The behaviour of these two classes of models is often similar.

3.3.1 Problems in existing models

We start with looking at the exponential network size growth $s = e^{\alpha t}$, where t is the physical time, and s is the system time. The advancing of the system time is driven by any event in the system. Since new edges can only be formed when new nodes arrive in the preferential

attachment models that we discuss, the network size is thus the same as the system time s .

However, the existing preferential attachment models that we reviewed in Section 3.1 do not distinguish the system time and the physical time. Thus, there is a natural gap between the observation and the models.

Before proceeding to more general considerations, we now use specifically the Bianconi-Barabási model [10] as an example, and show that the exponential network growth makes a significant difference to the node degree growth function. This can be generalised to other models similarly.

See Section 3.1.3 for details about the Bianconi-Barabási model.

Recall that the degree growth function in the Bianconi-Barabási model has been shown [10] to follow a power function

$$k_i(s, s_i, \eta_i) \sim \left(\frac{s}{s_i}\right)^{\beta(\eta_i)}, \quad (3.12)$$

where s_i is the system time (network size) when node i has appeared, s is the current system time, and the exponent β is a function of node fitness (for the basic model version, $\beta(\eta_i) \sim \eta_i$). As shown in Figure 3.11, such degree growth is clearly not invariant under the shift of the system time s .

However, if motivated by the exponential network growth size demonstrated in Figure 3.7, we assume that $s = e^{\alpha t}$, Equation (3.12) is then converted to

$$k_i(t, t_i, \eta_i) \sim e^{\alpha\beta(\eta_i)(t-t_i)}, \quad (3.34)$$

where t_i is the physical time when node i has appeared, and t is the physical observation time. Note this form is indeed time-invariant as it depends on the node age $\tau_i := t - t_i$ with no additional dependence on the node appearance time t_i .

We thus see that in the Bianconi-Barabási model, different network size growth functions lead to different node degree growth functions: When the number of nodes grows linearly with physical time as in the original model, it produces power-law degree growth function which is not time-invariant; when the number of nodes grows exponentially with physical time as in our empirical observation, it produces a time-invariant degree growth function which is of the exponential form.

There is, however, still an important difference between the growth produced by Equation (3.34) and the real data observations in Fig-

ure 3.6. While the former is of an exponential kind, the nearly linear curves in Figure 3.6 (log-log scale) suggest a power-law growth, much slower than the exponential growth. To resolve this disagreement, we proceed to more general preferential attachment models with fitness, ageing and exponential network growth, where the ageing effect causes a slowdown of the degree growth.

3.3.2 General preferential attachment with fitness, ageing and exponential network growth

The general model that we aim to study has three main contributing factors:

- *node degree* as a classical amplifier that can be introduced by various mechanisms such as the reference-copying process [38];
- *node fitness* as a reflection of intrinsic differences between the nodes;
- *ageing* as a mechanism that reflects the natural preference for new and, at the same time, limits the strong bias towards old nodes.

The probability that node i attracts a new link is usually assumed in the form

$$\Pi_i \sim k_i \eta_i R(\tau_i), \quad (3.35)$$

where τ_i is the age of node i in physical time, and $R(\tau_i)$ is typically a decreasing function which represents the gradual loss of the node's "relevance" and contributes to an eventual saturation of the degree growth. It is convenient to set $R(0) = 1$ so that ageing begins to influence the degree growth only later during each node's lifetime. Node fitness values are drawn from the distribution $\rho(\eta)$ which does not change with time. The number of nodes is assumed to grow exponentially with time, *i.e.*, $s = \exp(at)$.

The continuum approximation for the degree evolution [22] replaces the stochastic evolution of each node's degree with the average rate of its increase, $dk_i/ds = m\Pi_i(s)$ where m is the average number of new links created by a new node. Assuming that each new node creates *one link* to an already existing node, we obtain the differential equation of the degree growth:

$$\frac{dk_i}{ds} = \frac{k_i(s) \eta_i R(\tau_i)}{Z(s)} \quad (3.36)$$

Note the product of fitness and ageing has also been referred to as "relevance" in past literature [50].

where the normalisation term Z as a function of the network size s is $Z(s) := \sum_j k_j(s) \eta_j R(\tau_j)$. It is convenient here to switch to the physical time t , where the differential equation of the degree growth has the form

$$\frac{dk_i}{dt} = \alpha e^{\alpha t} \times \frac{k_i \eta_i R(\tau_i)}{Z(t)} \quad (3.37)$$

When t is large, the discrete sum in $Z(t)$ can be approximated with the double integral of the product $k_i \eta_i R(\tau_i)$ for all nodes, first over all possible node ages τ , then over all possible fitness values η ,

$$Z(t) \approx \int d\eta \rho(\eta) \eta \int_0^t d\tau k(\tau, \eta) R(\tau) \times \alpha e^{\alpha(t-\tau)}. \quad (3.38)$$

Since the network size grows exponentially, there are more nodes with smaller ages in the network, thus the density term $\alpha e^{\alpha(t-\tau)}$ is used when integrating over node ages τ . Denoting

$$\lim_{t \rightarrow \infty} \int d\eta \rho(\eta) \eta \int_0^t d\tau k(\tau, \eta) R(\tau) e^{-\alpha\tau} = \theta, \quad (3.39)$$

we can write the partition function as $Z(t) = \alpha e^{\alpha t} \theta$, *i.e.*, the partition function is proportional to the network size $e^{\alpha t}$, which then “happens” to cancel with the term in $ds/dt = \alpha e^{\alpha t}$. Equation (3.37) can thus be simplified to the form

$$\frac{dk_i}{d\tau_i} = \frac{k_i(\tau_i) \eta_i R(\tau_i)}{\theta}, \quad (3.40)$$

where we also use $\tau_i = t - t_i$ and replaced the derivative with respect to t by the derivative with respect to τ .

It is now convenient to define an auxiliary function $r(\tau)$ as the integral of the ageing function R :

$$r(\tau) := \int_0^\tau R(t) dt. \quad (3.41)$$

The solution of the differential equation, Equation (3.40), thus has the form

$$k_i(\tau_i, \eta_i) = \exp [\eta_i r(\tau_i) / \theta] \quad (3.42)$$

As we can see, different ageing functions now lead to different forms of the degree growth k . In particular, the ageing function $R(\tau) = (\tau + 1)^{-1}$ leads to a power-law degree growth $k(\tau, \eta) = \tau^{\eta/\theta}$ that can approximate the average degree growth in empirical data. In any case, Equation (3.42) shows that this model together with the assumption of an exponentially growing network size produces time-invariant degree growth.

After the term $\alpha e^{\alpha t}$ introduced in Equation (3.37) by the accelerating network growth being cancelled with the same term in $Z(t)$, the implied differential equation of the degree growth, Equation (3.40), is the same as when the uniform network growth ($s = t$) is assumed [50]. In contrast to [50] where the normalisation term $\sum_j k_j \eta_j R(\tau)$ converges only if $R(\tau)$ decays sufficiently fast (faster than $1/\tau$), we do not have a similar constraint here, as the exponential growth introduces the term $e^{-\alpha \tau}$ in θ ; this ensures convergence even when $R(\tau)$ decays no faster than $1/\tau$, for instance as in [71].

3.3.3 Ageing functions

In this subsection, we examine necessary properties of the ageing function $R(\tau)$, and discuss some functional forms of ageing functions.

Based on Equation (3.41) and (3.42), we can already spot some properties of the ageing function $R(\tau)$ and the corresponding auxiliary function $r(\tau)$. First, since the degree k cannot decrease in our model, $r(\tau)$ is a monotonically increasing function as opposite to $R(\tau)$. Second, given the initial value of node degree $k_i(0, \eta_i) = 1$, we have

$$r(0) = 0. \quad (3.43)$$

This helps to determine the exact form of $r(\tau)$ when $R(\tau)$ is given. Recalling $R(\tau)$ being a monotonic function that diminishes to 0 when $\tau \rightarrow +\infty$, and $R(0) = 1$, we indeed have a wide range of pairs of candidate functionals for $R(\tau)$ and $r(\tau)$.

In the following, we pick two pairs of $R(\tau)$ and $r(\tau)$ that lead to interesting results: One can recover the power-law degree growth curve as reported in the Barabási-Albert model and the Bianconi-Barabási model, and the other leads to the emergence of a sigmoid-curved degree growth function.

3.3.3.1 Ageing as power functions

We examine the situation when $R(\tau)$ is a power function, *i.e.*,

$$R(\tau) = (\tau + 1)^{-\gamma}, \quad \gamma > 0 \quad (3.44)$$

The term $+1$ guarantees $R(0) = 1$. To ensure that $R(\tau)$ is monotonically decreasing, γ must be greater than 0. This is different from [50] where γ must be greater than 1. We will explain why it is the case in

Section 3.4.2.3. As the first example, we will explain the situation of $\gamma = 1$ in most detail.

- The situation of $\gamma = 1$ which leads to $R(\tau) = (\tau + 1)^{-1}$ and $r(\tau) = \ln(\tau + 1)$ has been intensively studied in [71].

To begin with, we first recall and evaluate the denominator $Z(t)$ in Equation (3.37). When the age of the network t is large enough, we can apply the continuum approximation and get:

$$Z(t) = \sum_j k_j \eta_j R(\tau_j) \simeq \int d\eta \rho(\eta) \eta \int_0^t d\tau k(\tau, \eta) R(\tau) \alpha e^{\alpha(t-\tau)} \quad (3.45)$$

Again, the term $e^{\alpha(t-\tau)}$ indicates there are more nodes with smaller ages in the system, since the network size grows exponentially. Now let us plug in the ageing function $R(\tau) = (\tau + 1)^{-1}$, from which we can get the degree growth function according to Equation (3.42):

$$k_i(\tau_i, \eta_i) = (\tau_i + 1)^{\beta(\eta_i)}. \quad (3.46)$$

Here we recall $Z(t) = \alpha e^{\alpha t} \theta$ and denote the exponent as a function of η :

$$\beta(\eta_i) = \frac{\eta_i}{\theta} = \alpha e^{\alpha t} \frac{\eta_i}{Z(t)}. \quad (3.47)$$

The term $Z(t)$ is a double integral, first over the age of nodes, then over all fitness values. We first look at the inner integral which is over node ages for a certain fitness η :

$$\int_0^t d\tau k(\tau, \eta) R(\tau) \cdot \alpha e^{\alpha(t-\tau)} = \alpha e^{\alpha t} \int_0^t d\tau \frac{(\tau + 1)^{\beta(\eta)-1}}{e^{\alpha\tau}} \quad (3.48)$$

$$= \alpha e^{\alpha(t+1)} \cdot E_{1-\beta(\eta_i)}(\alpha), \quad (3.49)$$

where $E_{1-\beta(\eta_i)}(\alpha)$ is the generalised exponential integral function [57]. Thus, the term $Z(t)$ is proportional to the network size $s = e^{\alpha t}$ when $t \rightarrow +\infty$:

$$Z(t) \simeq \alpha e^{\alpha t} \theta, \quad (3.50)$$

with the constant

$$\theta = e^{\alpha} \cdot \int d\eta \rho(\eta) \eta \cdot E_{1-\beta(\eta_i)}(\alpha). \quad (3.51)$$

Plugging it in Equation (3.40), the analytically solution recovers the degree growth curve as a power function:

$$k_i(\tau_i, \eta_i) = (\tau_i + 1)^{\beta(\eta_i)} \quad (3.52)$$

given $\beta(\eta_i) \sim \eta_i$ (as in the Bianconi-Barabási model):

$$\beta(\eta_i) = \frac{\eta_i}{\theta}, \quad (3.53)$$

Thus we see that the model is self consistent.

- When $\gamma \neq 1$, we have

$$r(\tau) = \frac{(\tau + 1)^{1-\gamma}}{1-\gamma} + \zeta \quad (3.54)$$

Considering Equation (3.43), we can solve the constant of integration $\zeta = (\gamma - 1)^{-1}$. When $\gamma > 1$, we have $\limsup_{\tau \rightarrow +\infty} r(\tau) = \zeta$, *i.e.*, $r(\tau)$ is upper bounded, so is $k(\tau, \eta)$. When $0 < \gamma < 1$, both $r(\tau)$ and $k(\tau, \eta)$ do not converge when $\tau \rightarrow +\infty$.

To summarise, under power function ageing, the way for a node to achieve “immortality” (*i.e.*, its degree does not stop growing forever) is to have $\gamma \leq 1$ so that $\lim_{\tau \rightarrow +\infty} k(\tau, \eta)$ does not converge.

3.3.3.2 Ageing as exponential functions

We now examine the situation when $R(\tau)$ is an exponential function, *i.e.*,

$$R(\tau) = e^{-\gamma\tau}, \quad \gamma > 0 \quad (3.55)$$

To ensure that $R(\tau)$ is monotonically decreasing, γ must be greater than 0. Considering Equations (3.41) and (3.43), we have

$$r(\tau) = -\frac{e^{-\gamma\tau} - 1}{\gamma}, \quad (3.56)$$

which leads to:

$$k_i(\tau_i, \eta_i) = \exp\left(-\frac{\eta_i(e^{-\gamma\tau_i} - 1)}{\gamma\theta_{\text{exp}}}\right) \quad (3.57)$$

with the constant θ_{exp} satisfying

$$\theta_{\text{exp}} = \int d\eta \rho(\eta) \eta \cdot \int_0^{+\infty} \exp\left(-\frac{\eta(e^{-\gamma\tau} - 1)}{\gamma\theta_{\text{exp}}} - (\gamma + \alpha)\tau\right) d\tau \quad (3.58)$$

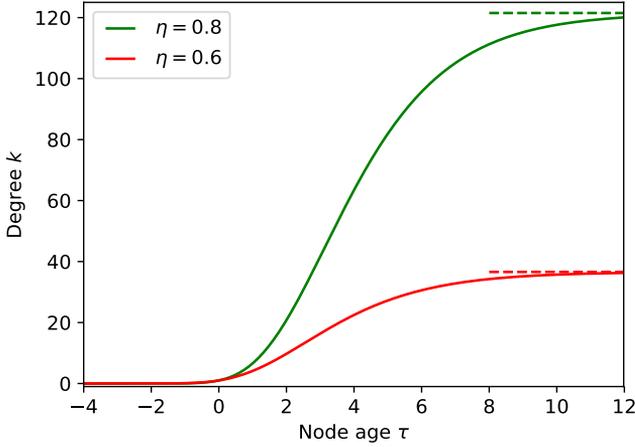


Figure 3.12: The theoretical degree growth curves under the exponential ageing $R(\tau) = e^{-\gamma\tau}$ with $\gamma = 0.5$ and $\theta_{\text{exp}} = 0.33$ as in Equation (3.57). The dashed lines show the corresponding theoretical asymptotes as in Equation (3.59). Note the X axis is extended to the negative domain in order to better illustrate the sigmoid shapes.

Equation (3.57) is a Gompertz function that has a sigmoid curve (as illustrated in Figure 3.12), and we have a converging degree growth function:

$$\lim_{\tau_i \rightarrow +\infty} k(\tau_i, \eta_i) = \exp\left(\frac{\eta_i}{\gamma\theta_{\text{exp}}}\right). \quad (3.59)$$

3.4 A NOVEL FORMALISM OF PREFERENTIAL ATTACHMENT MODELS

In Section 3.1 we have shown that the model introduced by Medo et al. in [50] produces time-invariant degree growth when the network size grows uniformly. In the last section, we have explained that the general preferential attachment models with fitness and ageing also produces time-invariant degree growth, when the network size grows exponentially.

One may naturally be curious now, about whether there exist other forms of the network size growth that are consistent with the time-invariant degree growth, and about the existence of a deeper relation-

ship between the time-invariant degree growth and the form of the network size growth.

We now proceed by showing that the uniform and exponential network growths are in fact the only two cases that are consistent with the time-invariant degree growth. To this end, we introduce a novel mathematical formalism for growing networks with the time-invariant degree growth as a fundamental assumption, but without an assumption on the network growth form in the first place.

To achieve a time-invariant degree growth for node attractiveness $\Pi_i \sim k_i \eta_i R(\tau_i)$, the differential equation of the degree growth function must take the form

$$\frac{dk_i}{d\tau_i} = ck_i \eta_i R(\tau_i) \quad (3.60)$$

where $c > 0$ is a positive constant. The resulting degree growth function is

$$k_i(\tau_i, \eta_i) = e^{c\eta_i r(\tau_i)} \quad (3.61)$$

where $r(\tau) := \int_0^\tau R(t) dt$. By recognizing $c = 1/\theta$, we recover Equation (3.42) as in the old formalism, hence the new formalism is consistent with the old one.

To avoid confusion, we use *degree growth function* or *degree growth* to refer to the function of k , and *degree increase* to refer to the first order derivative of k with respect to the physical time t .

Now, for a given fitness distribution $\rho(\eta)$, we introduce function $h(\tau)$ as the *average degree increase* of a node at age τ ,

$$h(\tau) = c \int d\eta \rho(\eta) \times \eta k(\tau, \eta) R(\tau). \quad (3.62)$$

We further introduce function $g(t)$ as the derivative of the network size s with respect to the physical time t , $g(t) = ds/dt$. Hence $g(t)$ is the rate at which new nodes arrive in the system. Since each node is assumed to create one link, $g(t)$ is also the total degree increase of all existing nodes at time t in physical time. Considering the asymptotic behaviour ($t \rightarrow \infty$) of the network growth, we can now write g as the convolution of h and g itself,

$$g(t) = \int_0^t h(\tau) g(t - \tau) d\tau. \quad (3.63)$$

Here we have the number of new links on the left side, and the same quantity, expressed through degree increase of all existing nodes, on the right side. Note so far we have not assumed any functional form of g .

Equation (3.63) is the core of our new formalism. It describes a linear time-invariant (LTI) system \mathcal{H} [33] whose impulse response function is h . Its input function happens to be the same as its output function,

$$g = \mathcal{H}g. \quad (3.64)$$

In other words, g is the eigenfunction of the LTI operator \mathcal{H} and thus it is of the exponential form $g(t) \sim e^{\sigma t}$ where $\sigma \in \mathbb{R}$, because g is a real function by definition. The eigenvalues of \mathcal{H} can be given by the Laplace transform of the impulse response h ,

$$\hat{h}(\sigma) = \mathcal{L}\{h(\tau)\} = \int_0^{+\infty} h(\tau)e^{-\sigma\tau} d\tau. \quad (3.65)$$

In Equation (3.64), the corresponding eigenvalue of g is exactly 1. We can thus get the parameter σ by solving

$$\hat{h}(\sigma) = 1. \quad (3.66)$$

The solution of σ determines the actual network growth.

- When $\sigma > 0$, we recover the exponential growth of network size $s(t) = e^{\sigma t}$ which was imposed by hand in the previous sections.
- When $\sigma = 0$, we have $g(t) = 1$ which implies the linear network growth $s(t) = t$ as in past literature [50].
- When $\sigma < 0$, the model is still in principle valid but outside the scope of this study, since it means that as time progresses, fewer and fewer nodes join the network.

We will explain the details of the impulse response function h and its Laplace transform, \hat{h} , for difference cases of preferential attachment in Section 3.4.2.

3.4.1 Estimation of the degree distribution

We now discuss how to estimate the degree distribution of the generated networks. The estimation result will then be used in the following sections to analyse the properties of our new formalism of the preferential attachment models, such as the relation to existing models, and to compare with numerical simulations.

See Section 2.1.5 for more details about LTI systems.

Let $P(K \geq k, t)$ denote the probability that a node has degree at least k at time t . Since the fitness distribution does not change with time, $P(K \geq k, t)$ can be written as

$$P(K \geq k, t) = \frac{\sum_{\eta} n(K \geq k, t, \eta)}{s} \quad (3.67)$$

where $n(K \geq k, t, \eta)$ represents the number of nodes with fitness η that have degree at least k at time t , and s is the network size which we assume to have the exponential form $s = e^{\alpha t}$.

Recalling Equation (3.61).

Since for a given fitness value η , the relation between k and τ is monotonous and independent of t , we can write τ as a function of k :

$$\tau(k, \eta) = r^{-1} \left(\frac{\log k}{c\eta} \right), \quad (3.68)$$

where r^{-1} is the inverse function of r .

See Section 3.1.2 for more details about the mean-field approximation.

Equation (3.67) can be then rewritten using the “mean-field” approximation [6, 12] as

$$P(K \geq k, t) = \frac{\int d\eta \rho(\eta) e^{\alpha \left[t - r^{-1} \left(\frac{\log k}{c\eta} \right) \right]}}{e^{\alpha t}}, \quad (3.69)$$

when the network is large enough, in particular in the limit $t \rightarrow \infty$. The only time-dependent term $e^{\alpha t}$ cancels out and we obtain

$$P(K \geq k) = \int d\eta \rho(\eta) e^{-\alpha r^{-1} \left(\frac{\log k}{c\eta} \right)}. \quad (3.70)$$

A stationary degree distribution $P(k)$ thus exists, *i.e.*,

$$P(k) \approx P(K \geq k) - P(K \geq k + 1). \quad (3.71)$$

3.4.2 Relation to existing preferential attachment models

We have explained our new formalism of preferential attachment network models, and its consistency with the exponential network growth. One might naturally ask, what is the relation between our new formalism and existing preferential attachment models? Are they compatible with each other, or is there an inherent difference? In this subsection, we show case by case, that starting from the new formalism, one can easily recover the existing models introduced in Section 3.1.

3.4.2.1 Relation to Barabási-Albert model

We now recall the Barabási-Albert model, in which the preferential attachment selection probability is proportional to the node degree:

$$\Pi_i \sim k_i. \quad (3.72)$$

DEGREE GROWTH Equation (3.72) means at any time, for any two existing nodes v_i and v_j in the network, the ratio of their degree increase must fulfil

$$\frac{dk_i}{dk_j} = \frac{k_i}{k_j}. \quad (3.73)$$

Moreover, in order to achieve a time-invariant degree growth for all nodes, the degree increase must be *autonomous*: It must involve only explicitly, the node age $\tau_i = t - t_i$ as measured in the physical time, but not directly t_i . Combining the two constraints, the only non-trivial degree increase must follow

$$\frac{dk_i}{d\tau_i} = ck_i, \quad (3.74)$$

where c is a positive constant that controls the rate of the degree increase. The resulting expected degree growth k_i as a function of τ_i , given the initial degree value $k_i(0) = 1$, is then

$$k_i(\tau_i) = e^{c\tau_i}. \quad (3.75)$$

Equation (3.75) is a simple exponential function, that seems different from Equation (3.5) that we had in the original Barabási-Albert model, which is a power function:

$$k_i(s, s_i) = \left(\frac{s}{s_i}\right)^{1/2}. \quad (3.5)$$

NETWORK GROWTH This seeming difference is due to the fact that, the two equations use different time: Equation (3.75), in our new formalism, uses the physical time t , while Equation (3.5), in the old formalism, uses the system time s . Yet, we have not established a relationship between the two kinds of time. For this, we will use the core equation of the new formalism:

$$g(t) = \int_0^t h(\tau)g(t-\tau)d\tau, \quad (3.63)$$

where $g(t)$ is the derivative of the network growth function:

$$g(t) = \frac{ds}{dt}, \quad (3.76)$$

and $h(\tau)$ is the average degree increase function. Since all nodes have the same degree growth function as in Equation (3.75), we have:

$$h(\tau) = \frac{dk}{d\tau} = ck = ce^{c\tau}. \quad (3.77)$$

Plugging in g and h into Equation (3.63) we get

$$g(t) = c \int_0^t e^{c\tau} g(t - \tau) d\tau, \quad (3.78)$$

and as usual, t has to be large enough in order for the continuum approximation to make sense. Solving Equation (3.78) we can see that

$$\frac{ds}{dt} = g(t) \sim e^{2ct} \quad (3.79)$$

when $t \rightarrow +\infty$. This means, if the network starts with $s(0) = 1$ node, the network size growth function should be

$$s(t) = e^{2ct}. \quad (3.80)$$

Here what is particularly interesting is that, the network growth $s = e^{2ct}$ can be considered as the sum of all nodes' degree growth $k = e^{ct}$ which join the network at different time, and they are both exponential functions. The exponent of the former ($2ct$) is twice as the one of the latter (ct). This actually explains the exponent $1/2$ in Equation (3.5).

By realizing $\alpha = 2c$, we naturally recover the exponential growth of the network size $s = e^{\alpha t}$ that we previously enforced in the model. This bridges the physical time t and the system time s with an exponential function. And if we plug it back, it is more clear that the two degree growth functions we discussed, Equations (3.75) and (3.5), are indeed equivalent.

DEGREE DISTRIBUTION For the stationary degree distribution, we can use the estimation in our formalism:

$$P(K \geq k) = \int d\eta \rho(\eta) e^{-\alpha r^{-1} \left(\frac{\log k}{c\eta} \right)}. \quad (3.70)$$

Specifically, there is no ageing effect in the Barabási-Albert model, so that the function r is the identity, *i.e.*, $r(x) = x$. The fitness distribution $\rho(\eta)$ can be seen as $\eta = 1$ for all nodes. Hence, we get the estimation of the complementary cumulative distribution function (CCDF) of the stationary degree distribution:

$$P(K \geq k) = k^{-2}, \quad (3.81)$$

which results in the probability density function:

$$p(k) = 2k^{-3}. \quad (3.82)$$

As we can see, the stationary degree distribution estimated with our formalism agrees with the existing result as in Equation (3.9) in Section 3.1.

SUMMARY We started with linear preferential attachment $\Pi_i \sim k_i$ without heterogeneous fitness and ageing as in the Barabási-Albert model, but from the perspective of our new formalism. Both formalisms reach the same conclusions of the degree growth and the stationary degree distribution of the Barabási-Albert model. However with our formalism, one can easily distinguish the system time s and the physical time t , and, starting from the time-invariant degree growth, one can recover the exponential network size growth $s(t) = e^{2ct}$ that is observed in the data.

One should notice that, in the case of the Barabási-Albert model, the parameter c can be seen as a scaler of the physical time, *i.e.*, $t' = ct$. Thus, c plays no role in the structure of the network. As we can see, the parameter c disappears in all network structure related results, such as the stationary degree distribution.

3.4.2.2 Relation to Bianconi-Barabási model

We now look at a more complicated case, the Bianconi-Barabási model, which we have discussed in Section 3.1.3. Compared with the Barabási-Albert model, the heterogeneous fitness is involved in the Bianconi-Barabási model. The preferential attachment selection probability Π_i of a node v_i is proportional to product of the node degree k_i and the node fitness η_i :

$$\Pi_i \sim k_i \eta_i. \quad (3.83)$$

The fitness distribution $\rho(\eta)$ is for all nodes that join the network at all time, and is assumed to be consistent over time.

DEGREE GROWTH This means, at any time, for any two existing nodes v_i and v_j in the network, the ratio of their degree increase must fulfil

$$\frac{dk_i}{dk_j} = \frac{k_i \eta_i}{k_j \eta_j} \quad (3.84)$$

Similar to the Barabási-Albert model case as we discussed in the previous subsection, in order to achieve a time-invariant degree growth for all nodes, the differential equation of the degree growth must be autonomous, *i.e.*, it depends only explicitly on the node age τ_i but not directly on t_i . Combining the two constraints, the only non-trivial degree increase must follow

$$\frac{dk_i}{d\tau_i} = ck_i \eta_i, \quad (3.85)$$

where c is a positive constant that controls the rate of the degree increase. We have explained in the previous subsection that the parameter c can be seen as a scaler of the physical time, and does not play a role in the structure of the generated network. The situation is the same here for the Bianconi-Bianconi model. Thus, for simplicity we use $c = 1$ in this subsection.

The resulting expected degree growth k_i as a function of τ_i , given the initial degree value $k_i(0) = 1$, is then

$$k_i(\tau_i) = e^{\eta_i \tau_i}. \quad (3.86)$$

NETWORK GROWTH Similar to the previous subsection, to bridge the exponential degree growth function as in Equation (3.86) to the seemingly different degree growth function as in Equation (3.12) that we had in the original Bianconi-Barabási model, which is a power function:

$$k_i(s, s_i, \eta_i) \sim \left(\frac{s}{s_i} \right)^{\beta(\eta_i)}, \quad (3.12)$$

with $\beta(\eta) = \frac{\eta}{c}$, we have to first bridge the system time s and the physical time t .

For this, we will use again the core equation of the new formalism:

$$g(t) = \int_0^t h(\tau)g(t-\tau)d\tau, \quad (3.63)$$

where $g(t) = ds/dt$ is the derivative of the network growth function, and the impulse response function $h(\tau)$ is the average degree increase function. However, at this point, since not all nodes have the same

degree increase function as in the Barabási-Albert model, we have to integrate over all possible η values to get $h(\tau)$:

$$h(\tau) = \int d\eta \rho(\eta) \eta e^{\eta\tau}. \quad (3.87)$$

Plugging in g and h into Equation (3.63) we get

$$g(t) = \int_0^t \int d\eta \rho(\eta) \eta e^{\eta\tau} g(t-\tau) d\tau, \quad (3.88)$$

and as usual, t has to be large enough in order for the continuum approximation to make sense.

Equation (3.88) does not have an obvious solution as it is the case for Equation (3.78) in the Barabási-Albert model. To solve it, it is necessary to get back to the core, Equation (3.63). We have explained in the beginning of this section that, Equation (3.63) describes a linear time-invariant (LTI) system \mathcal{H} [33] whose impulse response function is h :

$$g = \mathcal{H}g, \quad (3.64)$$

where the function g is the eigenfunction of the LTI operator \mathcal{H} . By using the LTI system, we can quickly jump into the conclusion that the real function g is the eigenfunction of the LTI operator \mathcal{H} . Thus, g is a real exponential function:

$$g(t) = Ae^{\sigma t} \quad (3.89)$$

where $A, \sigma \in \mathbb{R}$. It is now straightforward to get the network growth function

$$s(t) = e^{\sigma t}. \quad (3.90)$$

Till now, we have also recovered the exponential growth of the network size $s = e^{\alpha t}$ that we previously enforced in the Bianconi-Barabási model, if we realise $\alpha = \sigma$. Next, we discuss how to solve σ .

As we are now using the eigenvalue problem induced in the LTI system, and the eigenvalue λ in our problem is exactly 1 as shown in Equation (3.64), it is now natural to use the equation $\lambda = \hat{h}(\sigma) = 1$ in order to solve σ . For the Bianconi-Barabási model, the specific equation of the eigenvalue problem is

$$\hat{h}(\sigma) = \int_0^{+\infty} \int d\eta \rho(\eta) \eta e^{(\eta-\sigma)\tau} d\tau \quad (3.91)$$

$$= \int d\eta \rho(\eta) \int_0^{+\infty} \eta e^{(\eta-\sigma)\tau} d\tau = 1, \quad (3.92)$$

See Section 2.1.5 for details about LTI systems.

We see that in order for $\hat{h}(\sigma)$ to converge, $\eta - \sigma$ must be negative for any fitness value η . This makes sense in the scenario we want to model, because the degree growth of any node $e^{\eta t}$ must not surpass the growth of the entire network $e^{\sigma t}$. Thus we get

$$\int d\eta \rho(\eta) \frac{\eta}{\sigma - \eta} = 1 \quad (3.93)$$

The situation where Equation (3.93) lacks a solution will be discussed in Section 3.5.

for the general form of the Bianconi-Barabási model. The left hand side of Equation (3.93) is monotonic for σ , thus the equation should uniquely determine σ if such a solution exists.

Moreover, we can conclude that the exponent σ is positive from the negativity of $\eta - \sigma$, since all fitness η must be non-negative. This is significant, because it ensures that the network size $e^{\sigma t}$ grows with time.

This assumption does not lose generality.

As a special case, the Barabási-Albert model is equivalent to the case in the Bianconi-Barabási model when the fitness values of all nodes are the same. Again, for simplicity we assume all nodes have fitness 1. Equation (3.93) thus simplifies to

$$\frac{1}{\sigma_{\text{BA}} - 1} = 1, \quad (3.94)$$

which has the solution of $\sigma_{\text{BA}} = 2$. This is the same as we got in Equation (3.79).

This bridges the physical time t and the system time s with an exponential function $s = e^{\sigma t}$. And if we plug it back, it is more clear that the two degree growth functions we discussed, Equation (3.86) and (3.12), are indeed equivalent.

DEGREE DISTRIBUTION For the stationary degree distribution, we can as well use the estimation in our formalism:

$$P(K \geq k) = \int d\eta \rho(\eta) e^{-\alpha r^{-1} \left(\frac{\log k}{c\eta} \right)}. \quad (3.70)$$

Similar to the Barabási-Albert model, there is no ageing effect, so we get $r(x) = x$ as well here. Hence, we get the estimation of the complementary cumulative distribution function (CCDF) of the stationary degree distribution:

$$P(K \geq k) = \int d\eta \rho(\eta) k^{-\frac{\alpha}{\eta}}. \quad (3.95)$$

which results in the probability density function:

$$p(k) = \int d\eta \rho(\eta) \frac{\sigma}{\eta} k^{-\left(\frac{\sigma}{\eta}+1\right)}. \quad (3.96)$$

As we can see, the stationary degree distribution estimated with our formalism is a superposition of different power-law distributions. This agrees with the existing result as in Equation (3.17) in Section 3.1.3.

SUMMARY In this subsection, we analysed the Bianconi-Barabási model, *i.e.*, preferential attachment $\Pi_i \sim \eta_i k_i$ with heterogeneous fitness distribution $\rho(\eta)$ without ageing, from the perspective of our new formalism. Using our formalism, we recovered the same conclusions of the degree growth and the stationary degree distribution of the Bianconi-Barabási model. That is to say, our formalism is compatible with the Bianconi-Barabási model, but with exponential network growth.

3.4.2.3 Relation to the relevance decay model

Now we look into the relevance decay model as described in Section 3.1.5 but from the perspective of our new formalism. Compared with the Bianconi-Barabási model, an ageing function is involved to model the decaying attractiveness of nodes. It is thus a more general form of the preferential attachment network model compared with the Barabási-Albert model and Bianconi-Barabási model.

The preferential attachment selection probability Π_i of a node v_i is proportional to product of the node degree k_i , the node fitness η_i , and the ageing factor $R(\tau_i)$:

$$\Pi_i \sim k_i \eta_i R(\tau_i). \quad (3.97)$$

The fitness distribution $\rho(\eta)$ is for all nodes that join the network at all time, and is assumed to be consistent over time. The ageing factor $R(\tau)$ is a decreasing function of the node age τ .

DEGREE GROWTH At any time, for any two existing nodes v_i and v_j in the network, the ratio of their degree increase must fulfil

$$\frac{dk_i}{dk_j} = \frac{k_i \eta_i R(\tau_i)}{k_j \eta_j R(\tau_j)} \quad (3.98)$$

Similar to the Barabási-Albert model case as we discussed in the previous subsection, in order to achieve a time-invariant degree growth for all nodes, the degree increase must be autonomous, *i.e.*, it depends only explicitly on the node age τ_i but not directly on t_i . Combining the two constraints, the only non-trivial degree increase must follow Equation (3.60):

$$\frac{dk_i}{d\tau_i} = ck_i\eta_i R(\tau_i), \quad (3.60)$$

where c is a positive constant that controls the rate of the degree increase.

NETWORK GROWTH In the previous two subsections, we argued that the constant c merely acts as a scaler of the physical time, and does not play a role in the structure of the generated network in the Barabási-Albert model and Bianconi-Bianconi model. However, the situation is different in the relevance decay model, because the ageing function $R(\tau)$ is a function of the node age in physical time. Thus, different c can result in different ageing curves, thus resulting in different degree growth functions and network structures.

To be more clear, we examine the time-invariant degree growth function. Solving Equation (3.60) we get

$$k_i(\tau_i, \eta_i) = e^{c\eta_i r(\tau_i)}, \quad (3.61)$$

where the auxiliary function $r(\tau)$ is the integral of $R(\tau)$. In the exponent of the exponential function, the constant c is multiplied with $r(\tau)$ instead of the plain τ . Thus c is no longer a simple scaler of the physical time, unless $r(\tau) \equiv \tau$ as in the Barabási-Albert model and Bianconi-Barabási model.

In the original relevance decay model of Medo et al. [50] where the linear network growth $s = t$ is assumed, the degree growth function of a node resembles our Equation (3.61):

$$k_i = \exp\left(\frac{\eta_i}{\Omega^*} \int_0^{+\infty} R(\tau) d\tau\right), \quad (3.32)$$

where Ω^* is a constant when the network is large enough. It seems that here we do not need to use the exponential network growth to bridge the degree growth functions in our formalism and in the existing work, as we did in the previous two subsections with the Barabási-Albert model and Bianconi-Barabási model.

Before we try solving the paradox, we first into the network growth function in our formalism that is consistent with the time-invariant growth as in Equation (3.61).

For this, we will use again the core equation of the new formalism:

$$g(t) = \int_0^t h(\tau)g(t - \tau)d\tau, \quad (3.63)$$

where $g(t) = ds/dt$ is the derivative of the network growth function, and the impulse response function $h(\tau)$ is the average degree increase function.

We now write the average degree increase function as the impulse response function $h(\tau)$:

$$h(\tau) = c \int d\eta \rho(\eta) \eta k(\tau, \eta) R(\tau) \quad (3.62)$$

$$= c \int d\eta \rho(\eta) \eta e^{c\eta r(\tau)} R(\tau). \quad (3.99)$$

It is now important to solve $\hat{h}(\sigma) = 1$ to get the eigenvalue solution to the eigenproblem which describes the LTI system \mathcal{H} [33] whose impulse response function is h :

$$g = \mathcal{H}g. \quad (3.64)$$

Recalling the Barabási-Albert model and the Bianconi-Barabási model, where h is an exponential function (h_{BA}) or the superposition of multiple exponential functions (h_{fitness}):

$$h_{\text{BA}}(\tau) = ce^{c\tau}. \quad (3.77)$$

$$h_{\text{fitness}}(\tau) = \int d\eta \rho(\eta) \eta e^{\eta\tau}. \quad (3.87)$$

If we examine their Laplace transforms $\hat{h}_{\text{BA}} = \mathcal{L}\{h_{\text{BA}}\}$ and $\hat{h}_{\text{fitness}} = \mathcal{L}\{h_{\text{fitness}}\}$:

$$\hat{h}_{\text{BA}}(\tau) = c \int_0^{+\infty} e^{c\tau} e^{-\sigma\tau} d\tau. \quad (3.100)$$

$$\hat{h}_{\text{fitness}}(\tau) = \int_0^{+\infty} \int d\eta \rho(\eta) \eta e^{\eta\tau} e^{-\sigma\tau} d\tau. \quad (3.101)$$

See Section 2.1.4 for details about the Laplace transform and its region of convergence.

We can notice that the integrals h_{BA} and h_{fitness} over all positive values themselves diverge, thus the term $e^{-\sigma\tau}$ must be a decreasing function as a compensation, in order for the Laplace transform $\mathcal{L}\{h\} = \hat{h}(\sigma)$ to converge. We therefore see that σ has to be *at least* positive for the Barabási-Albert model and Bianconi-Barabási model.

However, this is not the case for the relevance decay model. The existence of the ageing function $R(\tau)$ slows down the degree growth, so that the latter grows slower than exponential. It is then possible that the integral of h over \mathbb{R} still converges without relying on the exponential compensation term $e^{-\sigma}$ with $\sigma > 0$, depending on how fast $R(\tau)$ decays.

- If $R(\tau)$ decays slow, $\int_0^{+\infty} \int d\eta \rho(\eta) \eta e^{c\eta r(\tau)} d\tau$ does not converge, then σ must be positive. An example of the ageing function for which σ must be positive is $R(\tau) = (\tau + 1)^{-\gamma}$ with $\gamma \leq 1$.
- If $R(\tau)$ decays fast, $\int_0^{+\infty} \int d\eta \rho(\eta) \eta e^{c\eta r(\tau)} d\tau$ converges, then σ can be positive, zero or negative, depending on the parameters c , γ and the fitness distribution $\rho(\eta)$. An example of the ageing function for which σ can be positive, zero or negative is $R(\tau) = (\tau + 1)^{-\gamma}$ with $\gamma > 1$.

The fact that σ can be non-positive is significant to determine the network growth and solve the paradox that we do not need to use the exponential network growth to bridge the degree growth functions in our formalism and in the original work of Medo et al.

Recall Equations (3.61) and (3.32).

Plugging in g and h into Equation (3.63) we get

$$g(t) = c \int_0^t \int d\eta \rho(\eta) \eta e^{c\eta r(\tau)} R(\tau) g(t - \tau) d\tau, \quad (3.102)$$

where the function g is the eigenfunction of the LTI operator \mathcal{H} . By using the LTI system, we still get conclusion that the real function g is the eigenfunction of the LTI operator \mathcal{H} , and has the form:

$$g(t) = Ae^{\sigma t} \quad (3.89)$$

where $A, \sigma \in \mathbb{R}$.

However, unlike in the Barabási-Albert model and Bianconi-Barabási model, we now do not have the constraint $\sigma > 0$. Hence, there can be the cases where $\hat{h}(\sigma) = 1$ results in $\sigma \leq 0$. Depending on σ , the network growth function $s(t)$ can be of different forms.

- If $\sigma > 0$, $g(t)$ is an exponentially growing function, so $s(t)$ is also an exponentially growing function:

$$s(t) = e^{\sigma t}. \quad (3.90)$$

This is similar as in the Barabási-Albert model and Bianconi-Barabási model with exponential network growth.

- If $\sigma = 0$, $g(t) = A$ is a constant, therefore $s(t)$ is a linear function, which resembles the original preferential attachment models where the linear network growth $s \sim t$ is assumed, and there is no need to distinguish the system time and the physical time.
- If $\sigma < 0$, $g(t)$ is an exponentially decaying function. A decaying $g(t)$ can model a dying community in which fewer and fewer nodes join the network as time passes.

SUMMARY To summarise, we have explained that in our new formalism of the preferential attachment models, as long as the ageing function $R(\tau)$ decays sufficiently fast such that $\int_0^{+\infty} \int d\eta \rho(\eta) \eta e^{c\eta\tau(\tau)} d\tau$ converges, the network growth can be linear as well. Hence, in this case there is no need to plug in an exponential network growth function to bridge the degree growth functions in our formalism and in the original work of Medo et al.

Recall Equations (3.61) and (3.32).

This also explains that, in the original work of Medo et al., the ageing must be faster than $(\tau + 1)^{-1}$ in order for their model to be self consistent: Because this is exactly the special case of our formalism where $\sigma = 0$, $\int_0^{+\infty} \int d\eta \rho(\eta) \eta e^{c\eta\tau(\tau)} d\tau$ has to converge.

3.5 BREAKING OF TIME-INVARIANCE

In Section 3.1.4 we have discussed the condensation phenomenon that arises in preferential attachment network models and the so called winner-takes-all effect. In this section, we aim at explaining different condensation phenomenon using our new formalism of preferential attachment network models. We show that the condensation phenomenon corresponds to the breaking of the time-invariance in our new formalism, and, the time-invariance of the system as a whole is broken when $\hat{h}(\sigma) = 1$ does not have a solution.

3.5.1 *Breaking of time-invariance in Bianconi-Barabási model*

To explain this, we start with the new formalism of the Bianconi-Barabási model [10] where $\Pi_i \sim \eta_i k_i$ and no ageing is present. The time-invariant degree growth function is thus a special case of Equation (3.61) where $r(\tau) \equiv \tau$, *i.e.*,

$$k_i(\tau_i, \eta_i) = e^{c\eta_i \tau_i}. \quad (3.103)$$

One can realise that the constant c is merely a time scaler and is free of choice here, so for simplicity we let $c = 1$. The impulse response can be written as

$$h(\tau) = \int d\eta \rho(\eta) \eta e^{\eta \tau}. \quad (3.104)$$

Solving

$$\hat{h}(\sigma) = \mathcal{L}\{h(\tau)\} = \int d\eta \rho(\eta) \frac{\eta}{\sigma - \eta} = 1 \quad (3.105)$$

gives us the exponential growth rate of the network σ . Since the degree growth rate of every node must not surpass the growth rate of the entire network, we have an additional constraint $\sigma \geq \eta_{\max}$ where η_{\max} is the maximum fitness.

Since $\hat{h}(\sigma)$ is a decreasing function of σ , the maximum value of $\hat{h}(\sigma)$ is achieved at $\sigma = \eta_{\max}$. However, for some fitness distributions, $\hat{h}(\eta_{\max})$ is still smaller than 1 which is required by Equation (3.66), and consequently, $\hat{h}(\sigma) = 1$ does not have a solution. When such fitness distributions are taken, the node with the leading fitness will eventually attract almost all edges (a “winner-takes-all” effect). The network growth is thus asymptotically approached by the maximum degree growth, *i.e.*, $g \sim k_{\max}$ and, in the case of the Bianconi-Barabási model, $g(t) \sim e^{\eta_{\max} t}$. This situation has been intensively studied in [9], where the authors have approached the problem using the formalism used to study the Bose-Einstein condensation.

An example of the fitness distribution that can lead to the Bose-Einstein condensation is

$$\rho(\eta) = (\lambda + 1)(1 - \eta)^\lambda \quad (3.106)$$

where $\eta \in [0, 1]$ and λ is a parameter that controls how skewed the distribution is. Note $\rho(\eta)$ is a probability density function, thus

$$\int_0^1 \rho(\eta) d\eta = 1. \quad (3.107)$$

When $\lambda \leq 1$, we can find a solution of σ that is larger than the largest possible η , *i.e.*, $\eta_{\max} = 1$. The critical parameter $\lambda_{\text{BE}} = 1$ that leads to the condensation can be obtained. When $\lambda > 1$, we fail to find a solution of σ that is larger than $\eta_{\max} = 1$. In fact, by mapping fitness η to energy ϵ at temperature T with $\eta = e^{-\epsilon/T}$, one can realise that our Equation (3.105) is equivalent to Equation (10) in [9].

Figure 3.13 shows how λ affects the skewness of $\rho(\eta)$ and σ . As we can see from Figure 3.13a, the distribution of η becomes more skewed when λ is larger. In Figure 3.13 we can see that when $\lambda \leq \lambda_{\text{BE}} = 1$, the exponential growth rate σ of the network is a decreasing function of λ . However, the numerical solver cannot find a satisfying σ when $\lambda > \lambda_{\text{BE}} = 1$. This is aligned with our analytical analysis.

3.5.2 Breaking of time-invariance in superlinear preferential attachment

With our new formalism, we can also address other cases in which a similar condensation phenomenon arises, for instance the superlinear preferential attachment $\Pi_i \sim k_i^\gamma$ with $\gamma > 1$, where eventually a single node connects to nearly all other nodes [40]. This can be seen from the fact that the time-invariant degree growth function

$$k(\tau) = [(1 - \gamma)c\tau + 1]^{1/(1-\gamma)}, \quad (3.108)$$

resulting from the differential equation $dk/d\tau = ck^\gamma$ with $\gamma > 1$ and $k(0) = 1$, displays a finite-time divergence at $\tau = [c(\gamma - 1)]^{-1}$. As a result, the Laplace transform of the average degree growth function $h(\tau)$,

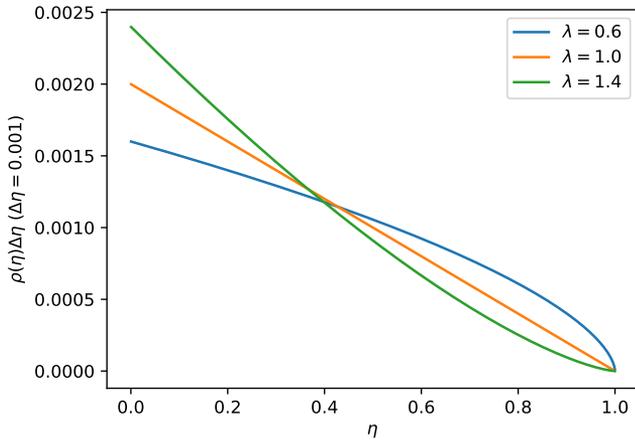
$$\hat{h}(\sigma) = \int_0^\infty k(\tau)e^{-\sigma\tau}d\tau, \quad (3.109)$$

does not converge for any real value σ , because the exponential term $e^{-\sigma\tau}$ cannot counterbalance the growth of k which is not exponentially bounded. Thus we see that $\hat{h}(\sigma) = 1$ lacks a solution in \mathbb{R} .

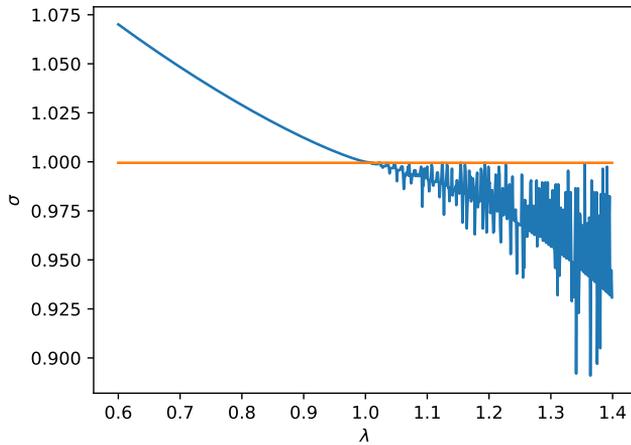
3.5.3 No condensation in the presence of ageing

In this subsection, we examine the situation when the ageing function $R(\tau)$ is present. We see that similar breaking of the time-invariance does not occur under a mild condition, that is if we assume the ageing function diminishes to 0 when we take the time limit $\tau \rightarrow \infty$, *i.e.*,

$$\lim_{\tau \rightarrow \infty} R(\tau) = 0. \quad (3.110)$$



(a)



(b)

Figure 3.13: Subfigure (a) shows the fitness distribution $\rho(\eta) = (\lambda + 1)(1 - \eta)^\lambda$ with different three parameters λ . When λ is larger, the fitness distribution is more skewed. Subfigure (b) shows the corresponding σ as a function of λ solved using a numerical solver, which fails to find solutions when $\lambda > 1$.

This is valid for all ageing functions that have been discussed in Section 3.3.3, for instance power functions and exponential functions.

To prove this, we first examine the convergence of $\hat{h}(\sigma)$,

$$\hat{h}(\sigma) = \mathcal{L}\{h(\tau)\} = \mathcal{L}\left\{c \int d\eta \rho(\eta) \eta k(\tau, \eta) R(\tau)\right\}. \quad (3.111)$$

We now use the linearity of the Laplace transform \mathcal{L} , and rewrite the equation above as

$$\hat{h}(\sigma) = c \int d\eta \rho(\eta) \eta \mathcal{L}\{k(\tau, \eta) R(\tau)\}. \quad (3.112)$$

Hence, $\hat{h}(\sigma)$ converges if $\mathcal{L}\{k(\tau, \eta) R(\tau)\}$ converges for all η .

Since

- $k(\tau, \eta_{\max}) \geq k(\tau, \eta)$ for all η , where η_{\max} is the maximum fitness, *i.e.*, the degree growth of the node with the maximum fitness is faster than other nodes;
- the ageing function $R(\tau)$ is a monotonically decreasing function,

it is natural now to see that, $\hat{h}(\sigma)$ converges if $\mathcal{L}\{k(\tau, \eta)\}$ converges for η_{\max} , with η_{\max} being the maximum fitness.

Recalling Equation (3.61), we have

$$\mathcal{L}\{k(\tau, \eta_{\max})\} = \int_0^\infty e^{c\eta_{\max}r(\tau)} e^{-\sigma\tau} d\tau. \quad (3.113)$$

We thus examine the ratio

$$\frac{e^{c\eta_{\max}r(\tau+1)-\sigma(\tau+1)}}{e^{c\eta_{\max}r(\tau)-\sigma\tau}} = e^{c\eta_{\max}(r(\tau+1)-r(\tau))} \cdot e^{-\sigma}. \quad (3.114)$$

When taking the limit $\tau \rightarrow \infty$, since $r'(\tau) = R(\tau)$ and $\lim_{\tau \rightarrow \infty} R(\tau) = 0$, we see that $e^{c\eta_{\max}(r(\tau+1)-r(\tau))}$ approaches 1. Therefore the examined ratio is less than 1 when $\sigma > 0$, which guarantees the convergence of $\mathcal{L}\{k(\tau, \eta_{\max})\}$ and, consequently, of $\hat{h}(\sigma)$. Since $\hat{h}(\sigma)$ is a continuous monotonic function, there is always one solution of σ that makes $\hat{h}(\sigma) = 1$.

See Section 2.1.4 for details about the linearity of the Laplace transform.

3.6 EVALUATION

In order to systematically validate our analysis and the new formalism proposed in the previous sections, in this section, we simulate the generation process of our model to grow synthetic networks, and compare the result with the analytical prediction.

3.6.1 Methodology

We now explain the spirit and details of the methodology we use in the simulations.

In order to evaluate the new formalism of preferential attachment models we proposed, we do not directly control the network size growth. Instead, we enforce the time-invariant degree growth of the nodes according to our model. The growth curve of the network is left to be observed and compared with the model's analytical prediction.

Synthetic networks have initially a small number N_0 of nodes with degree one each. In our experiments we have found that different values for N_0 does not influence the results significantly.

In the simulation process, time runs in short time steps of size Δt . Since our model assumes continuous time, having a large Δt may distort the actual model behaviour, because then a large number of links can be formed in a single time step, and the network evolves visibly stepwise. Thus, to limit the effects of time discretisation, Δt has to be small enough. However, a too small Δt leads to a lot of computational overhead, and the simulations take too long to finish. In our experiments we have found that $\Delta t = 0.02$ with

$$e = \lim_{n \rightarrow +\infty} \left(1 + \frac{1}{n}\right)^n$$

$$(1 + \Delta t)^{1/\Delta t} = 2.70 \simeq e \quad (3.115)$$

is a good balance between the precision of the simulation and the computational overhead.

The degree increase of nodes in our model is characterised by the differential equation as in Equation (3.60):

$$\frac{dk_i}{d\tau_i} = ck_i\eta_i R(\tau_i) \quad (3.60)$$

However, if at each discrete time step we take

$$\Delta k_i = ck_i\eta_i R(\tau_i)\Delta t \quad (3.116)$$

as the degree increase of a node, the node degrees will then not be guaranteed to have realistic integer values, and the network evolution will be deterministic which contradicts the reality.

In our experiments instead, at each discrete time step, for each existing node v_i in the network, we drawn from the Poisson distribution with the mean value $\Delta k_i = ck_i\eta_i R(\tau_i)\Delta t$ as the increase of the node degree. This makes the simulation of the node degree increase equivalent to the Poisson process where the expected increase is Δk_i in the time interval Δt . In this way, node degrees will always have realistic integer values rather than floating point values which represent the expected degrees, such as in Figure 3.1 and Figure 3.3.

A new node with degree one is added to the network whenever the degree of an existing node is increased by one. In this way, we effectively imply the time-invariance of the degree growth, and have the possibility to observe the emergent network growth.

3.6.2 Evaluation focusses

We evaluate our new formalism with different parameters, and focus on the following four major aspects in the evaluation.

1. Network growth.

We predict with our new formalism that, when there is no network condensation, the time-invariant degree growth is consistent with the exponential network growth with the exponent σ as the solution of Equation (3.66):

$$\hat{h}(\sigma) = 1, \quad (3.66)$$

where \hat{h} is the Laplace transform of the average degree increase of all nodes.

Moreover, the asymptotic network growth will be approached by the maximum degree growth in the network when the network condensation happens.

2. Degree distribution.

We plot the degree distributions of the synthetic networks produced by the model, and compare with the analytical prediction given by the mean-field estimation as in Section 3.4.1. In the simulation, the estimation of the cumulative degree distribution can be given numerically with Equation (3.70):

$$P(K \geq k) = \int d\eta \rho(\eta) e^{-ar^{-1}\left(\frac{\log k}{c\eta}\right)}, \quad (3.70)$$

given parameters c , $\rho(\eta)$ and $r(\tau)$. The stationary degree distribution function $P(k)$ can also be given numerically:

$$P(k) \approx P(K \geq k) - P(K \geq k + 1). \quad (3.71)$$

Note the mean-field estimation behaves less accurate on high degrees when the distribution shows an exponential cut-off (see Section 3.1.2).

3. Degree growth.

Our new formalism is built on the time-invariance of the degree growth. Although the time-invariance of the degree growth is implied in the simulation process, we still want to observe if it is actually the case.

Moreover, the analytical degree growth function as given in our new formalism is

$$k_i(\tau_i, \eta_i) = e^{c\eta_i r(\tau_i)}. \quad (3.61)$$

To compare the actual degree growth curves, we also record the degrees of certain nodes in the synthetic network at different times. Since the degree growth of individual nodes can be largely influenced by the randomness of the above-mentioned Poisson process, we group the nodes and observe the average growth of their degrees. The details of how we pick and group nodes can be found in the corresponding experiments below.

4. Network condensation.

We predict with our new formalism that the condensation in networks (the winner-takes-all effect) takes place when Equation (3.66) lacks a solution. For example, this happens in the superlinear preferential attachment, and in the Bianconi-Barabási model when the fitness distribution is skewed. Moreover, the ageing effect can prevent the winner-takes-all effect when the preferential attachment is linear, despite having a skewed fitness distribution. Network condensation can be observed from

the degree distribution and the asymptotic maximum degree growth.

3.6.3 Results

In this subsection, we illustrate the evaluation results of our framework concerning the four major aspects mentioned above, with different settings of the time-invariant degree growths. We provide more supplementary results in Appendix A.

3.6.3.1 Results for Barabási-Albert model

We now show the experimental results of the synthetic networks generated from the linear preferential attachment model without heterogeneous fitness and ageing (as in the Barabási-Albert model) using our framework:

$$\frac{dk_i}{d\tau_i} \sim k_i. \quad (3.117)$$

Figure 3.14 shows the network growth curves. Figure 3.15 shows the degree distributions. Figure 3.16 shows the average degree growth curves of nodes.

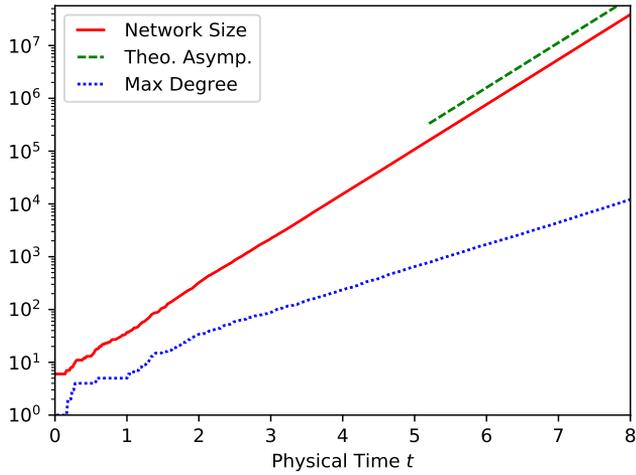


Figure 3.14: The network growth curve (red) of a synthetic network generated from the linear preferential attachment ($dk/d\tau \sim k$ as in the Barabási-Albert model) using our framework. The slope of the green dashed line indicates the theoretical estimation of the asymptotic network growth. The blue dashed curve shows the growth of the maximum degree in the network. Note that the plot uses the linear-log scale to illustrate the exponential curves.

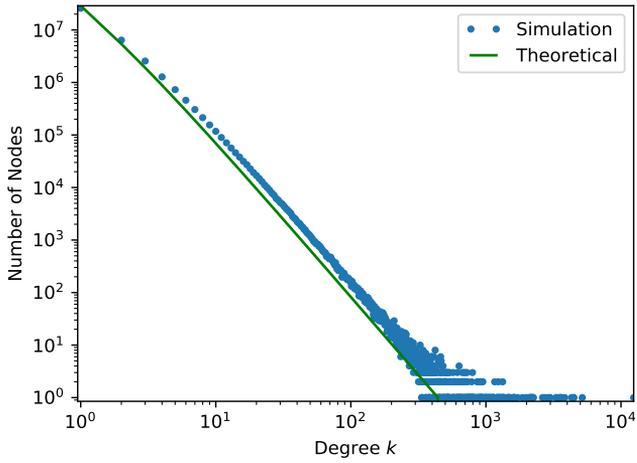


Figure 3.15: The degree distribution of a synthetic network generated from the linear preferential attachment ($dk/d\tau \sim k$ as in the Barabási-Albert model) using our framework. The green curves indicate the theoretical estimation of the degree distributions. Note that the plot uses the log-log scale to illustrate the scale-free distributions.

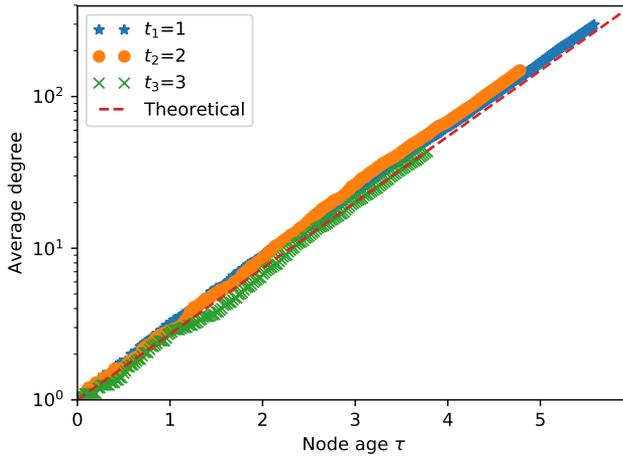


Figure 3.16: The average degree growth curves of nodes with different ages in a synthetic network generated from the linear preferential attachment ($dk/d\tau \sim k$ as in the Barabási-Albert model) using our framework. The slope of the red dashed line indicates the theoretical estimation of the degree growth. Note that the plot uses the linear-log scale to illustrate the exponential curves.

3.6.3.2 Results for Bianconi-Barabási model

We now show the experimental results of the synthetic networks generated from the preferential attachment model with heterogeneous fitness, without ageing (as in the Bianconi-Barabási model) using our framework:

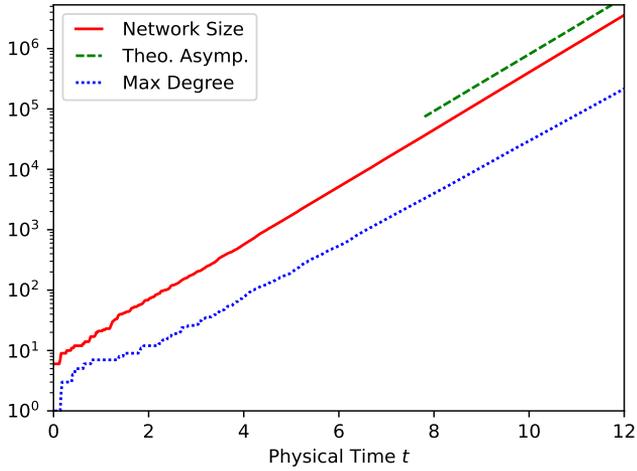
$$\frac{dk_i}{d\tau_i} \sim k_i \eta_i, \quad (3.118)$$

with the fitness distribution

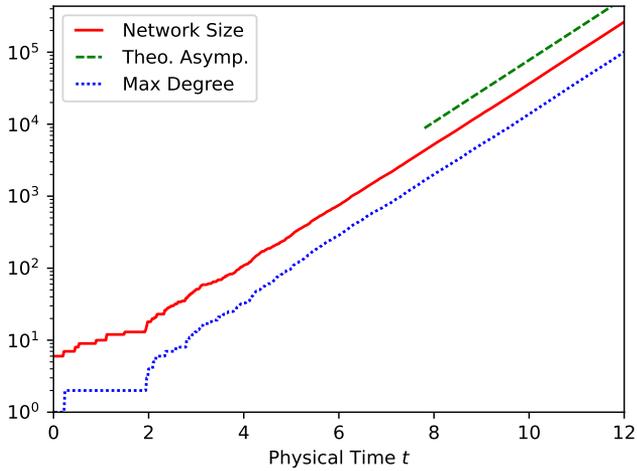
$$\rho(\eta) \sim (1 - \eta)^\lambda, \quad (3.119)$$

where λ is the parameter to be varied.

Figure 3.17 shows the network growth curves for $\lambda = 0.5$ and $\lambda = 1.5$. Figure 3.18 shows the degree distributions for $\lambda = 0.5$ and $\lambda = 1.5$. Figure 3.19 shows the average degree growth curves of nodes.



(a) $\lambda = 0.5$



(b) $\lambda = 1.5$

Figure 3.17: The network growth curves (red) of synthetic networks generated using our framework with preferential attachment with heterogeneous fitness ($\Pi \sim k\eta$ as in the Bianconi-Barabási model). The green dashed lines indicate the asymptotic slopes of the theoretical estimation of the network growth. The blue dashed curves show the growth of the maximum degree in the networks. Note that the plots use the linear-log scale.

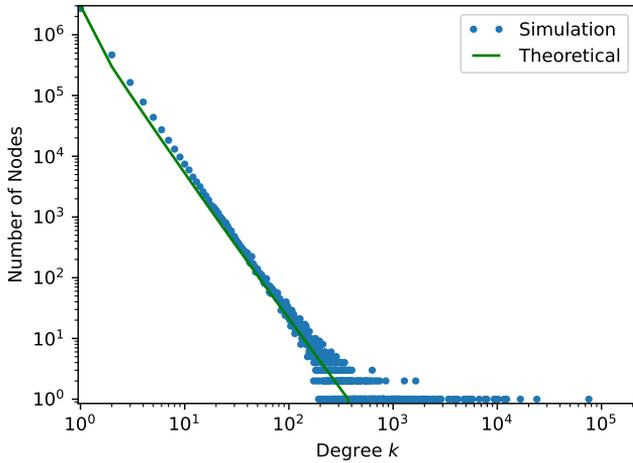
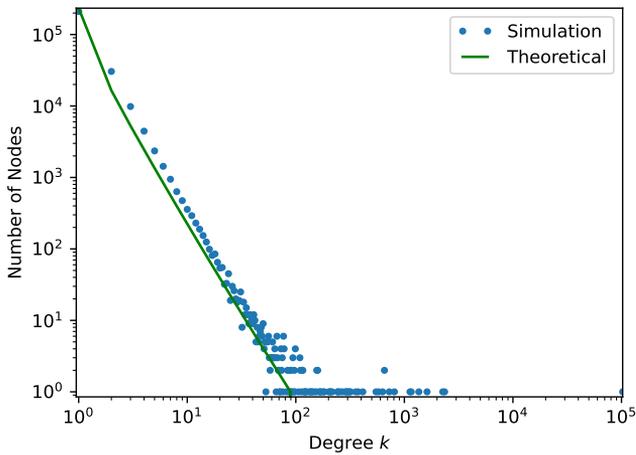
(a) $\lambda = 0.5$ (b) $\lambda = 1.5$

Figure 3.18: The degree distributions of the synthetic networks generated using our framework with preferential attachment with heterogeneous fitness ($\Pi \sim k\eta$ as in the Bianconi-Barabási model). The green lines show the theoretical estimation. Note that the plots use the log-log scale.

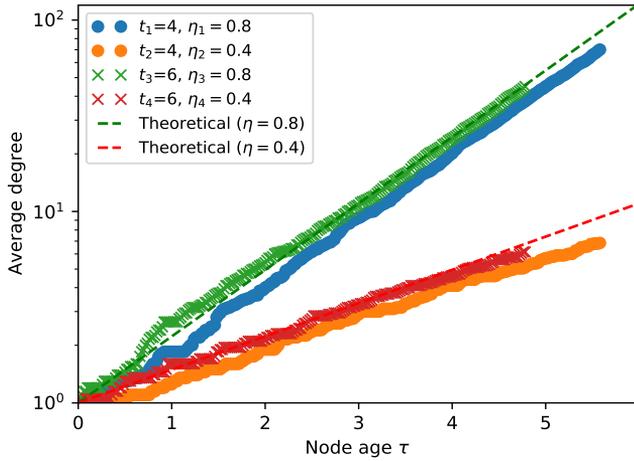


Figure 3.19: The average degree growth curves of four groups of nodes with different time when joining the network and different fitness values. The synthetic network is generated using our framework with preferential attachment with heterogeneous fitness ($\Pi \sim k\eta$ as in the Bianconi-Barabási model). The dashed lines show the theoretical estimation. Note that the plot uses the linear-log scale.

3.6.3.3 Results with ageing effect

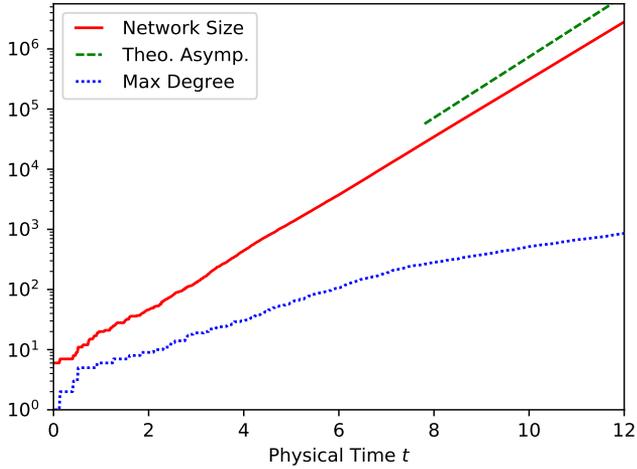
We now show the experimental results of the synthetic networks generated from the preferential attachment model with heterogeneous fitness and ageing effect (as in the relevance decay model) using our framework:

$$\frac{dk_i}{d\tau_i} \sim k_i \eta_i R(\tau), \quad (3.120)$$

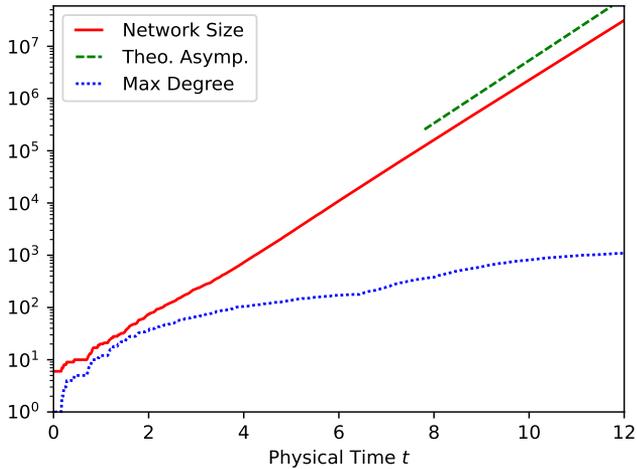
with the fitness distribution

$$\rho(\eta) \sim (1 - \eta)^\lambda, \quad (3.121)$$

with $\lambda = 1.5$ and the ageing function $R(\tau)$. We evaluate the power and exponential ageing as discussed in Section 3.3.3.



(a) $R(\tau) = (\tau + 1)^{-1}$



(b) $R(\tau) = e^{-0.5\tau}$

Figure 3.20: The network growth curves (red) of synthetic networks generated using our framework with preferential attachment with heterogeneous fitness and (a) ageing as power function; (b) ageing as exponential function. ($\Pi \sim k\eta R(\tau)$ as in the relevance decay model). The green dashed lines indicate the asymptotic slopes of the theoretical estimation of the network growth. The blue dashed curves show the growth of the maximum degree in the networks. Note that the plots use the linear-log scale.

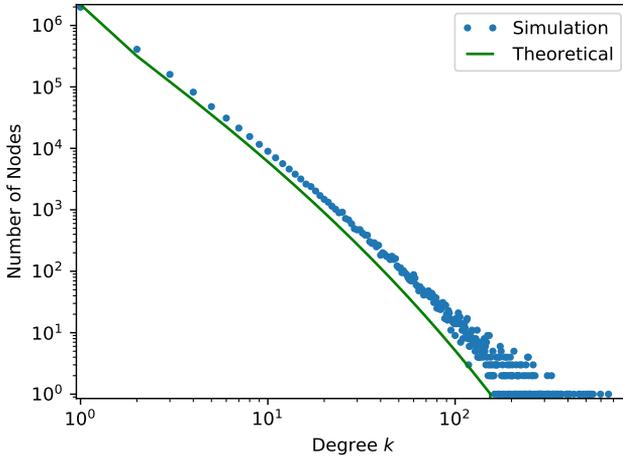
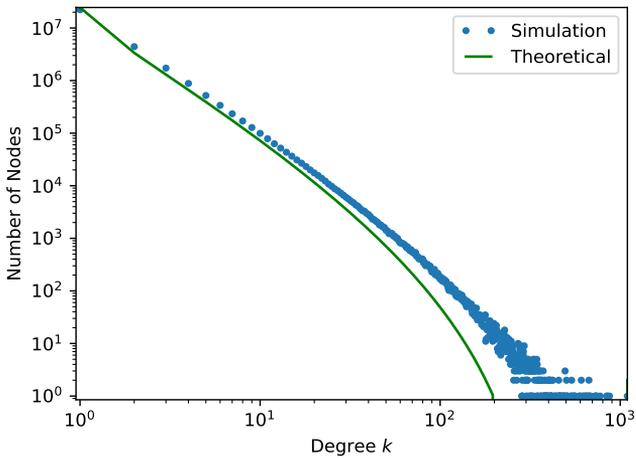
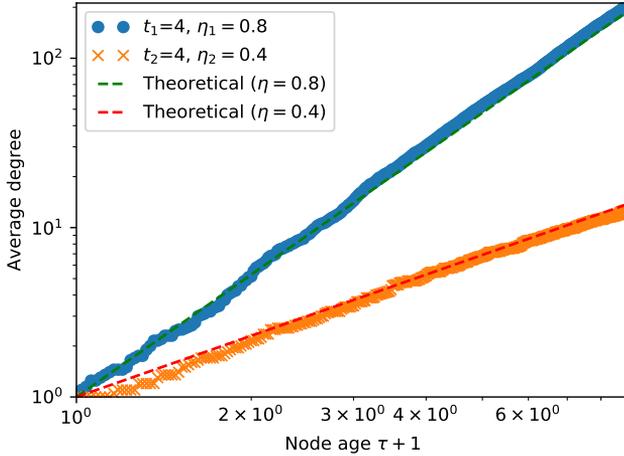
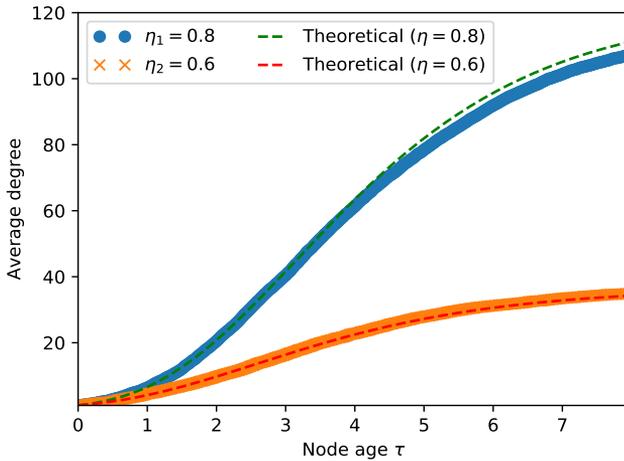
(a) $R(\tau) = (\tau + 1)^{-1}$ (b) $R(\tau) = e^{-0.5\tau}$

Figure 3.21: The degree distributions of the synthetic networks generated using our framework with preferential attachment with heterogeneous fitness and (a) ageing as power function; (b) ageing as exponential function. ($\Pi \sim k\eta R(\tau)$ as in the relevance decay model). The green lines show the theoretical estimation. Note that the plots use the log-log scale.



(a) $R(\tau) = (\tau + 1)^{-1}$



(b) $R(\tau) = e^{-0.5\tau}$

Figure 3.22: The average degree growth curves of groups of nodes with approximately the same age and different fitness values. The synthetic network is generated using our framework with preferential attachment with heterogeneous fitness and (a) ageing as power function; (b) ageing as exponential function. ($\Pi \sim k\eta R(\tau)$ as in the relevance decay model). The dashed lines show the theoretical estimation. Note that (a) uses the log-log scale, while (b) uses the linear-linear scale.

3.6.3.4 Results with superlinear preferential attachment

We now show the experimental results of the synthetic networks generated from the superlinear preferential attachment model without heterogeneous fitness and ageing using our framework:

$$\frac{dk_i}{d\tau_i} \sim k_i^\gamma. \quad (3.122)$$

In our experiment we take $\gamma = 1.2$.

Figure 3.23 shows the network growth curves. Figure 3.24 shows the degree distributions. Figure 3.25 shows the average degree growth curves of nodes.

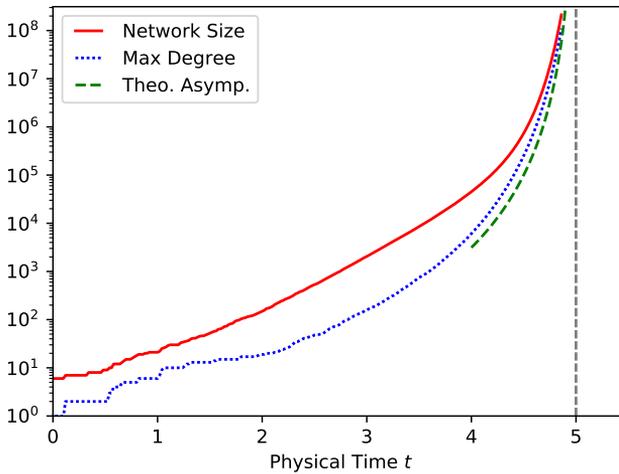


Figure 3.23: The network growth curve (red) of a synthetic network generated from the superlinear preferential attachment ($dk/d\tau \sim k^{1.2}$) using our framework. The green dashed line indicates the theoretical estimation of the asymptotic network growth, where the network size should go to infinity at $t = 5$ (indicated with the vertical grey dashed line). The blue dashed curve shows the growth of the maximum degree in the network. Note that the plot uses the linear-log scale.

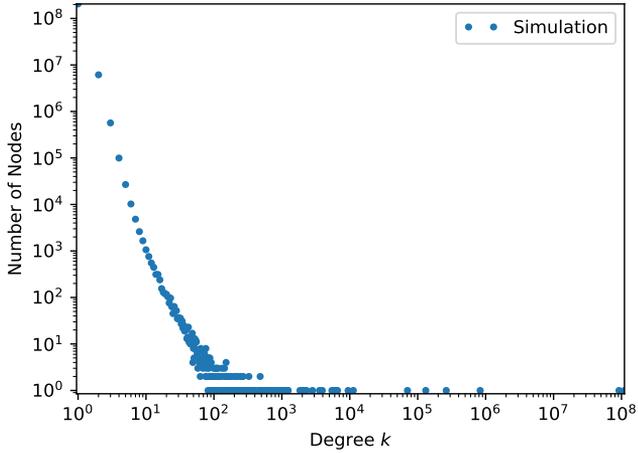


Figure 3.24: The degree distribution of a synthetic network generated from the superlinear preferential attachment ($dk/d\tau \sim k^{1.2}$) using our framework. Note that the plot uses the log-log scale.

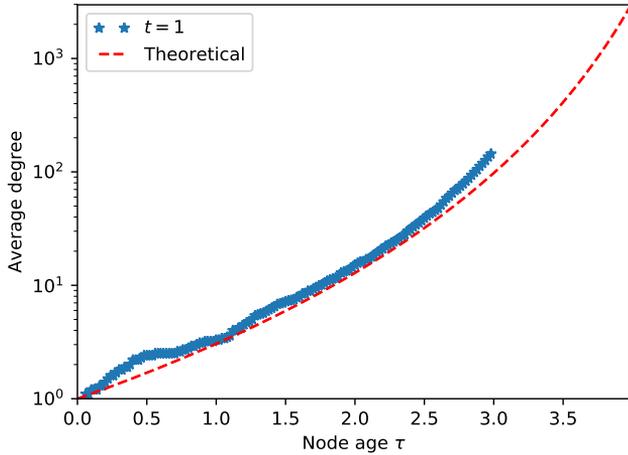


Figure 3.25: The average degree growth curve of a group of 10 nodes which joined the network at approximately the same time $t = 1$. The synthetic network is generated from the superlinear preferential attachment ($dk/d\tau \sim k^{1.2}$) using our framework. The red dashed line indicates the theoretical estimation of the degree growth. Note that the plot uses the linear-log scale.

3.6.4 Result analysis

We now analyse the findings in our experimental results.

3.6.4.1 Evaluation of the network growth

Simulation results shown in the last subsection demonstrate that the emerging network growth in all cases eventually matches the theoretical prediction.

When Equation (3.66) has a solution σ , the network size exhibits an exponential growth with the exponent σ (Figure 3.14, 3.17a, 3.20a, 3.20b).

In the case of the Bose-Einstein condensation, the network size grows with the exponent η_{\max} (Figure 3.17b). Figure 3.25 shows the superlinear preferential attachment which results in a network growth that is not exponentially bounded and can be approximated by the theoretical maximum degree growth curve $k = (5/(5-t))^5$ which follows from Equation (3.108) for $\gamma = 1.2$. As a result, the network size approaches infinity when $t = 5$ (indicated with the vertical dashed line).

3.6.4.2 Evaluation of the degree distributions

For the degree distributions, our theoretical results based on Equation (3.70) match the synthetic results well, especially in the low degree domain.

Depending on the parameters, some networks have power-law shaped, well-defined long tail degree distributions (Figure 3.15, 3.18a, 3.21a), while some exhibits an exponential cut-off (Figure 3.21b), where the simulation result deviates from the theoretical estimation in high degree domain (similar to Figure 3.2b).

For the superlinear preferential attachment (Figure 3.24), since the network growth is not exponential, Equation (3.70) does not apply and no stationary degree distribution is shown. This is in line with the known conclusion that superlinear preferential attachment networks lack an asymptotic stationary degree distribution [40].

In the case of the Bose-Einstein condensation (Figure 3.18b), there are some clear “winners” with large degree values, yet the slope of the

estimated degree distribution still matches the simulation result for low degrees.

3.6.4.3 *Evaluation of the degree growth*

Our experiment code is programmed with the implication that the degree growth should be time-invariant, which is the assumption of our new formalism.

To verify if the simulation achieves the time-invariance, we plot Figure 3.16 and 3.19. The overlapping curves of the average degree growth of nodes with the same time joining the network show that the degree growth in the simulation is indeed time-invariant as intended.

Moreover, further plots show our theoretical estimation of the degree growth function matches with the simulation results. Figure 3.16 and 3.19 exhibit exponential degree growth as expected, for the linear preferential attachment with or without heterogeneous fitness, but without ageing. Figure 3.22a verifies the degree growth when we have the ageing function $R(\tau) = (\tau + 1)^{-1}$ has the form of the power function, which exhibits as straight lines in the log-log plot.

Figure 3.22b verifies the degree growth when we have exponential ageing function is part of the sigmoid curve in the linear-linear plot. In Figure 3.25 the degree growth for the superlinear preferential attachment is not exponentially bounded, as we have expected.

3.6.4.4 *Evaluation of the network condensation*

Apart from the extreme values shown in the degree distribution, another important signature of the winner-takes-all effect is that the maximum degree eventually dominates the network growth, taking a fixed fraction of the network size.

In our new formalism, this happens when Equation (3.66) lacks a solution, in the case of the superlinear preferential attachment (Figure 3.23) as well as in the Bianconi-Barabási model when the Bose-Einstein condensation occurs [9], *i.e.*, $\lambda \geq \lambda_{\text{BE}} = 1$ (Figure 3.17b).

As we have proven, in the presence of a diminishing ageing function, there can be no winner-takes-all effect although $\lambda > \lambda_{\text{BE}}$. This can be verified in Figure 3.20a for the power ageing, and Figure 3.20b for the exponential ageing, where the maximum degree takes a smaller

and smaller fraction of the network size, albeit the network having a skewed fitness distribution.

3.7 SUMMARY

In this chapter, we provide a comprehensive analysis of various existing preferential attachment models, and various social effects in the resulting networks. We have found that the accelerated growth of the network, albeit common in real systems, is typically neglected when analysing network growth models. We find that instead of being an unnecessary nuisance, the form of the network growth is an important component which together with preferential attachment, fitness, and ageing shapes the network.

Building on the observation that the average node degree growth in two different citation networks is time-invariant, we formulate a new analytical formalism which allows us to take the time-invariance of the degree growth as the first principle and study the emerging network properties. The time-invariance of the degree growth is a natural property in networks that eventually reach “stationary” growth: Their old and new nodes are alike in the way how their degree grows and saturates.

We use the new formalism to show that only two forms of network growth are compatible with the time-invariant degree growth: A uniform growth that is assumed by most network models, and an exponential growth that is often found in real data. The simultaneous presence of time-invariant degree growth and an exponential network growth can be thus seen as empirical confirmation of these two patterns being self-consistent in growing networks with preferential attachment. The new formalism naturally connects various network condensation phenomena that have been previously studied separately: The Bose-Einstein condensation in the Bianconi-Barabási model with skewed fitness distributions, and the condensation in the superlinear preferential attachment model. We also prove that ageing is necessary to reproduce realistic degree growth curves, and can prevent the network condensation with mild conditions.

LEARNING ACROSS NETWORKS

In this chapter, we describe our work on the topic of learning across networks. We specifically address the problem of predicting user labels in online social networks with transfer learning, based on known user labels in other online social networks.

In Section 4.1, we introduce related works on transfer learning, transfer learning for network analysis and user study.

In Section 4.2, we demonstrate our transfer learning-based label prediction approach TraNet in detail. TraNet is a three-level approach, based on

- the extraction of structural features of nodes,
- the transformation of the extracted features to a common feature space,
- the classification of nodes based on the transformed structural features.

To help understanding TraNet, we also use the role transfer task on the Wikipedia user interaction network data as an run-through example.

In Section 4.3, we show the experimental evaluation of the role transfer task in the Wikipedia user interaction networks. We also show another use case of TraNet, which is identifying trusted users in ARIS Community¹, the online platform for customers of Software AG, the second-largest German software vendor. We also compare the performance of TraNet with other approaches.

In Section 4.4, we summarise the chapter.

¹ <https://www.ariscommunity.com>

4.1 RELATED WORK

In this section, we introduce the concept of transfer learning, as well as existing transfer learning algorithms for network analysis and user study in networks.

4.1.1 *Transfer learning*

Traditionally, if we perform machine learning tasks such as classification, regression and clustering [80], one assumption is made: The training data from which we learn, and the test data to which we want to apply the knowledge we learnt are sampled from the same domain, *i.e.*, they are assumed to have the same feature distribution [61]. However, this assumption might not stand true in many real scenarios, especially when we have completely new domains to work on, or completely new tasks to accomplish. In this case, transferring knowledge learnt from an existing domain is necessary. This concept of knowledge transfer is called *transfer learning*. In this dissertation, we adopt the definition of transfer learning given by Pan and Yang [61]:

“Given a source domain D_S and learning task T_S , a target domain D_T and learning task T_T , transfer learning aims to help improve the learning of the target predictive function $f_T(\cdot)$ in D_T using the knowledge in D_S and T_S , where $D_S \neq D_T$ or $T_S \neq T_T$.”

We use the term *source dataset* to refer to a dataset that belongs to the source domain D_S , that we learn knowledge from; and *target dataset* to refer to a dataset that belongs to the target domain D_T , to which we want to transfer the knowledge which we have learnt from the source dataset.

In our study, both source and target datasets contain different user interaction networks, where each node represents a user, and each edge represents a user interaction or relation. We assume the ground truth about user labels is present in the source dataset, but not in the target dataset.

4.1.2 *Transductive transfer learning*

In [61], the term *transductive transfer learning* is defined as a special case of transfer learning, where, in contrast to inductive learning, *transductive* emphasises that the learning targets T_T and T_S are the same:

“Given a source domain D_S and a corresponding learning task T_S , a target domain D_T and a corresponding learning task T_T , transductive transfer learning aims to improve the learning of the target predictive function $f_T(\cdot)$ in D_T using the knowledge in D_S and T_S , where $D_S \neq D_T$ and $T_S = T_T$. In addition, some unlabelled target-domain data must be available at training time.”

Note that the terms *transductive learning* and *transfer learning* are orthogonal, *i.e.*, a transductive learning problem can be either a transfer learning problem or a non-transfer learning problem, and vice versa [4].

In our study, we do not want to limit our method to the cases where the target dataset is present during training. In other words, the target dataset can be completely unseen during the training part (see Section 4.2), therefore no target-domain data (labelled or unlabelled) are available at training time. This makes the task more challenging [4].

To summarise, the domains D_S and D_T are two different social networks. The learning tasks T_S and T_T are the same: both to determine user labels in the networks. The predictive function $f_T(\cdot)$ gets the structural features of a node as input, and returns the labels of this node. Therefore, the task we focus on can be categorised as transductive transfer learning, if we relax the condition that “some unlabelled target-domain data must be available at training time.”

4.1.3 *Transfer learning algorithms for network analysis*

Henderson et al. have proposed ReFeX (Recursive Feature eXtraction) [32], an algorithm to extract nodes’ structural features in a network. The idea of ReFeX is that an actor in a network is not only characterised by who the actor is, but also who its neighbours are, and where it is located in the network. Thus, ReFeX recursively combines nodes’ local features and neighbourhood features. Evaluation has shown that ReFeX is scalable and suitable for a variety of

transfer learning network analysis tasks, *e.g.*, node classification and de-anonymisation.

In [31], Henderson et al. describe an algorithm called RolX to extract node roles in networks automatically. RolX first performs an unsupervised soft clustering for all nodes in the network using matrix decomposition (*e.g.*, non-negative matrix factorisation [42]) on the structural features extracted by ReFeX or other algorithms. The result of the matrix decomposition, a soft clustering of nodes, is regarded as a probability-based role membership assignment of the nodes.

RolX also supports transfer learning (across-network role classification), if ground truth of user roles is present in one network (the source dataset). The probability-based role memberships of the nodes in the source dataset can serve as features to train a classifier (*e.g.*, a logistic regression model), which can then be used to classify user roles in another network, based on the role membership features in the target dataset obtained in the same way.

4.2 PROPOSED METHOD

In this section, we illustrate the detailed procedure of our transfer learning-based approach for cross-network user study. To better explain our approach, we use the user role transfer task on the Wiki-talk datasets as a running example.

EXAMPLE The Wiki-talk dataset is a set of user interaction networks in different languages of Wikipedia, where each directed edge

$$(\text{User_ID_A}, \text{User_ID_B}, \text{timestamp})$$

represents a user interaction: User A wrote a message on User B's talk page at a certain time.

Each Wikipedia user has an access level [79], which we interpret as the following roles:

- **Administrator.** Administrators refer to the accounts that have high level of access to contents and maintenance tools in Wikipedia.
- **Bot.** Bots are used in Wikipedia for automatically or semi-automatically improving contents.

See Appendix C for a more detailed description of the Wiki-talk dataset.

- **Normal user.** Other users that are not categorised as administrators or bots.

The proportions of both bots and administrators vary highly among all sub-datasets, from 0.0027% to 5.97% and from 0% to 0.72% respectively.

The user role transfer task assumes the complete knowledge of user roles in the source network, and tries to predict the unknown user roles in the target network.

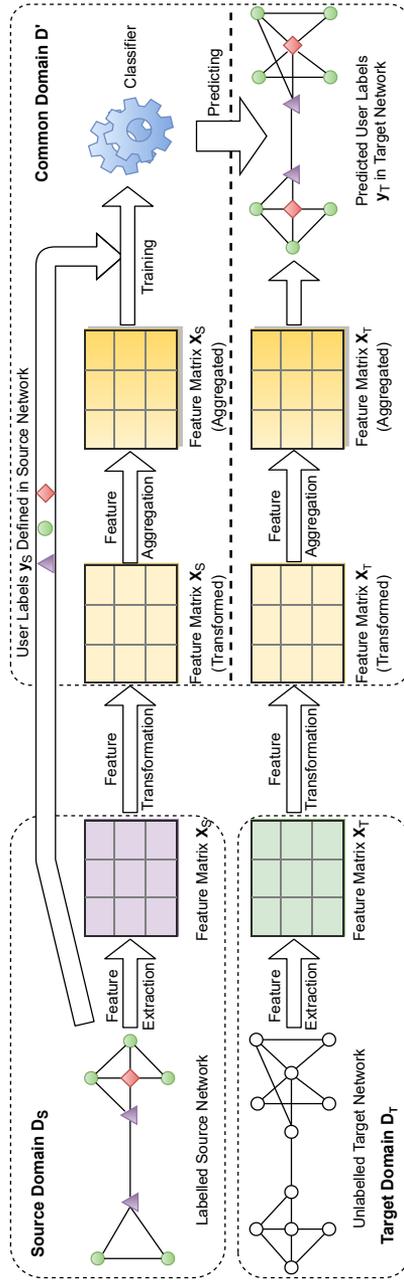


Figure 4.1: Overview of TraNet, the transfer learning approach for user study described in this chapter. We learn from the source dataset how to detect the following three (exemplary) predefined user labels: central users (diamonds), bridging users (triangles), and normal users (circles); and transfer the knowledge to the unlabelled target network to predict the user labels in it. The components within the source domain D_S and within the common domain D' but above the dashed line are obtained during *training*, while the ones within the target domain D_T and within the common domain D' but below the dashed line are obtained during *application*.

4.2.1 Overview

Figure 4.1 shows the overview of the transfer learning procedure of TraNet, which can be divided into two parts: *training* (top row) and *application* (bottom row).

4.2.1.1 Feature matrix

In both training and application parts, we generate structural features of nodes from the corresponding network (the source network G_S for the training part, and the target network G_T for the application part). The feature generation processes for both parts can be done separately, because we do not assume the availability of the target dataset during the training part. In each part, with a social network G as input, we extract the feature matrix $\mathbf{X}_{n \times dim}$, where n is the number of nodes in G , and dim is the dimension of the structural features.

Inside $\mathbf{X}_{n \times dim}$, each node (user) has dim structural features \mathbf{x}_j with $j \in \{1, 2, \dots, dim\}$, which are expected to be non-domain-specific. We have the following three steps to get the feature matrix \mathbf{X} :

- base feature extraction;
- feature transformation;
- feature aggregation.

They will be explained respectively in the following subsections.

4.2.1.2 User labels

In the source dataset, we also have the ground truth of user labels in the network during training, which is denoted as a vector \mathbf{y}_S of length n_S , where n_S is the size (*i.e.*, number of nodes) of the source network. Each value $\mathbf{y}_i \in \{1, 2, \dots, l\}$ with $i \in \{1, 2, \dots, n_S\}$ in \mathbf{y}_S denotes the label of the i th user, where l is the total number of possible user labels.

With \mathbf{y}_S and the feature matrices $\mathbf{X}_S, \mathbf{X}_T$ of both source and target networks, the user label prediction problem reduces to a classification problem in machine learning.

EXAMPLE In the Wiki-talk dataset we use, we have the following three roles: “Administrator”, “Bot” and “Normal user”, thus y can be set to have three values: $y \in \{1, 2, 3\}$, with each representing a role respectively. However, if we wish to build binary classifiers for “Administrator” and “Bot”, then we can as well set y to be binary for each classifier. For instance for the Administrator classifier y can take the value in $\{1, 2\}$, and 1 represents an Administrator, 2 a non-Administrator. In our implementation we choose the latter approach, in order to analyse the performance of the classifier for each role.

4.2.2 Base feature extraction

In social networks, users behave in different ways. A user’s behaviour is reflected in her surrounding network structure, and thus we can examine the structural features of a node to study the user. For instance, users with similar social behaviour can be classified into the same *role*. Thus, we can examine the common neighbourhood patterns of nodes with the same role to build a structural profile of this role, and use it to identify the roles of other users with their extracted structural features. [31].

Structural features of nodes can be extracted by looking at only the structure (*e.g.*, the adjacency matrix) of the network, without requiring information on additional attributes of nodes or links (*e.g.*, users’ geolocation as node attributes, or message contents as link attributes in a user interaction network) [32]. In traditional machine learning, it is often helpful to take this additional information into consideration. However, these node and link attributes are usually domain-specific, and may not be applicable in other domains. In transfer learning, blindly transferring knowledge may not be successful, or even make the performance of learning worse [61]. On the contrary, structural features are usually common across networks. Therefore, in this study we only consider structural features.

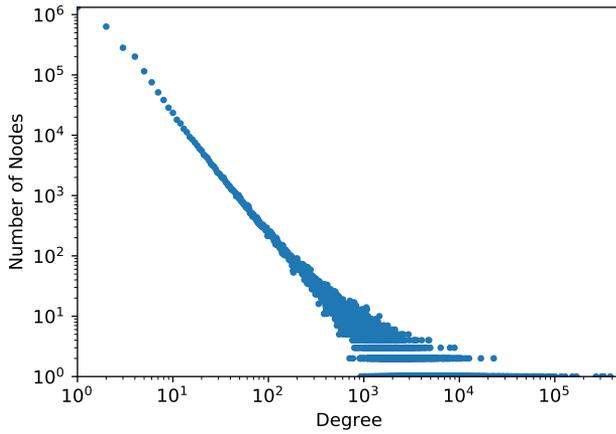
Given the adjacency matrix \mathbf{A} of a network, we compute the following five structural features for each node as its *base features*:

- k : The degree of a node;
- k^{in} : The indegree of a node;
- k^{out} : The outdegree of a node;
- c^{core} : The coreness of a node;

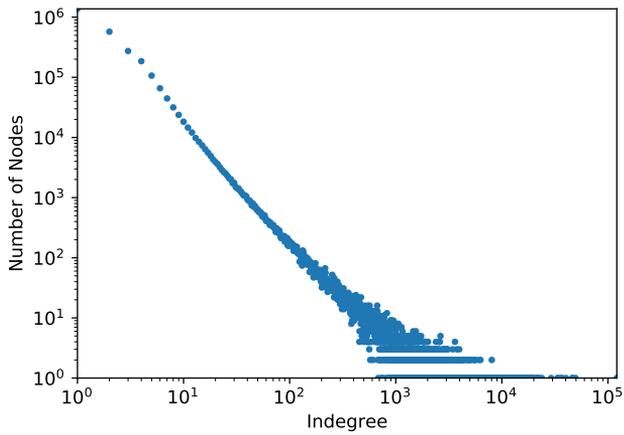
See Section 2.5 for structural properties of nodes.

- C^{local} : The local clustering coefficient of a node [77]. We ignore edge directions when we compute the local clustering coefficient for each node;
- pr : The PageRank of a node [59].

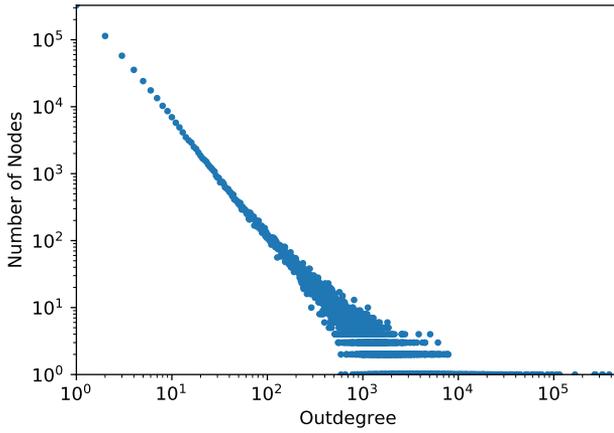
EXAMPLE In Figure B.5 we show the base feature distributions of nodes in the Wiki-talk-en network. The feature distributions of the other Wiki-talk networks are shown in Appendix B.



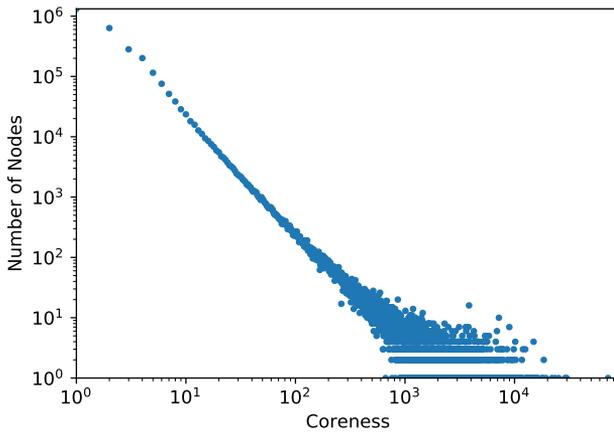
(a) Degree distribution



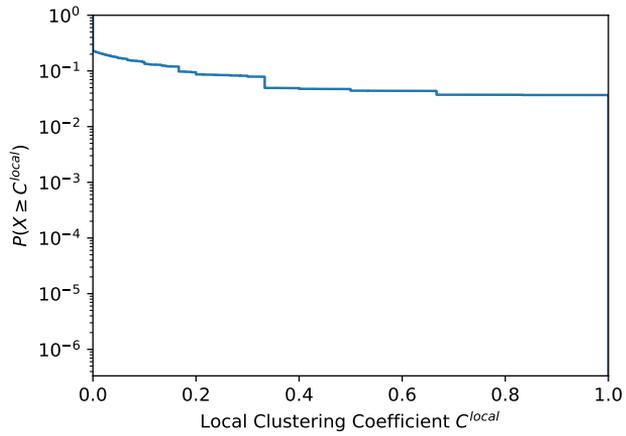
(b) Indegree distribution



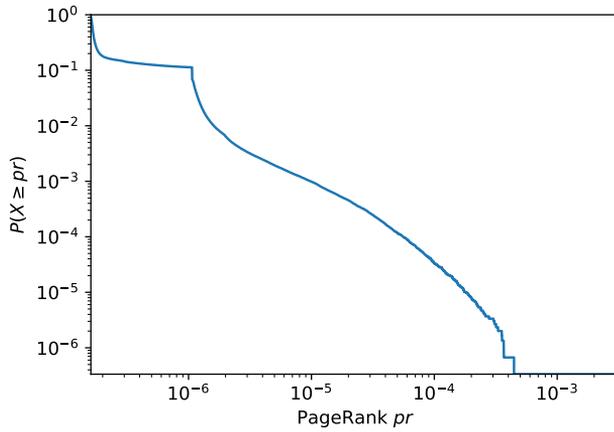
(c) Outdegree distribution



(d) Coreness distribution



(e) Local clustering coefficient distribution



(f) PageRank distribution

Figure 4.2: The base feature distribution of the Wiki-talk-en network.

4.2.3 Feature transformation

The main challenge in transfer learning is that the distributions of features differ between the source and target datasets. Thus, features that are extracted from different networks are often not directly comparable. Therefore, after all base features are extracted, we transform them via different methods in order to make them comparable across networks. The transformation of features to a dataset-independent space of values is performed separately for each dataset.

Feature transformation is especially difficult since the target dataset might not be seen during the training phase [4, 39]. In our approach, the general idea of feature transformation is to define a common feature distribution for each kind of base feature, which is more likely to be comparable across networks. Therefore, the feature transformation procedure is network-independent and order-free, *i.e.*, we do not need to access the target network when we perform feature transformation for the source network, and vice versa.

In the following, we discuss the different transformation methods we use in TraNet.

4.2.3.1 Quantile transformation

The quantile transformation transforms any given feature value into its *quantile* value [65], which is always within the range $[0, 1)$. The quantile of a feature value is defined as the probability that any value in this domain is less than this value:

$$x' = \text{quantile}(x) = P(x_j < x), j \in \{1, 2, \dots, |\mathbf{x}|\}, \quad (4.1)$$

where x' serves as the transformed feature value. For example, if the original feature values are:

$$\mathbf{x} = [1, 0, 1, 5, 2] \quad (4.2)$$

then the transformed feature values x' (quantiles) will be:

$$\mathbf{x}' = [0.2, 0, 0.2, 0.8, 0.6] \quad (4.3)$$

Considering the definition of quantile, the transformed features always have a value within $[0, 1)$, and thus they are comparable across feature domains. In our implementation, we use the quantile transformation for the base feature local clustering coefficient.

However, using the quantile transformation will lose information of the original feature's distribution. Here we consider a simple example: A different set of original feature values $[1, 0, 1, 5, 4]$ will also lead to the same quantiles $[0.2, 0, 0.2, 0.8, 0.6]$ as in the previous example. More generally, x' will be uniformly distributed within $[0, 1]$, if there are no equal values in the original feature values x , regardless of the distribution in x .

4.2.3.2 Power-law transformation

See Section 2.4.2 for details about power-law degree distributions.

Many real world networks exhibit power-law alike degree distributions [18, 52], whose analytical form can be approximated by the following equation:

$$p(x) = c \cdot x^{-\alpha}. \quad (4.4)$$

In Chapter 3 we have intensively studied the generative mechanics where power-law alike degree distributions can emerge, including preferential attachment, heterogeneous fitness, and ageing.

For some other base features such as the coreness, our empirical observations also show that they follow power-law distribution approximately in real social networks (Figure B.5d).

Now, assume a base feature x follows power-law approximately, using this prior knowledge, we can plug in the idea of quantile transformation as in Equation (4.5), and transform x into one common power-law distribution:

$$\int_{x_{\min}}^x p(x)dx = \int_{x'_{\min}}^{x'} p'(x')dx' \quad (4.5)$$

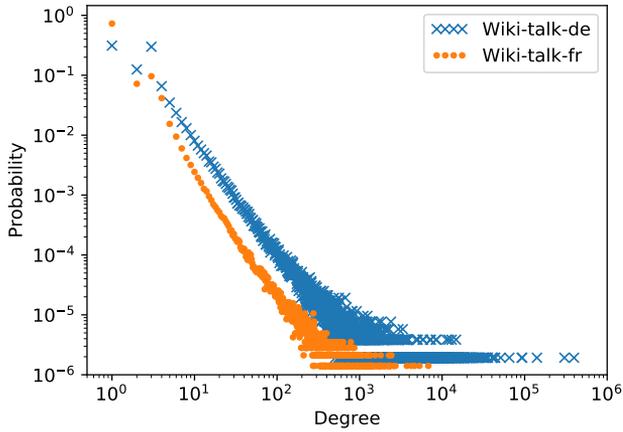
We choose the power-law distribution:

$$p'(x') = x'^{-2} \quad (x' \in [1, +\infty)) \quad (4.6)$$

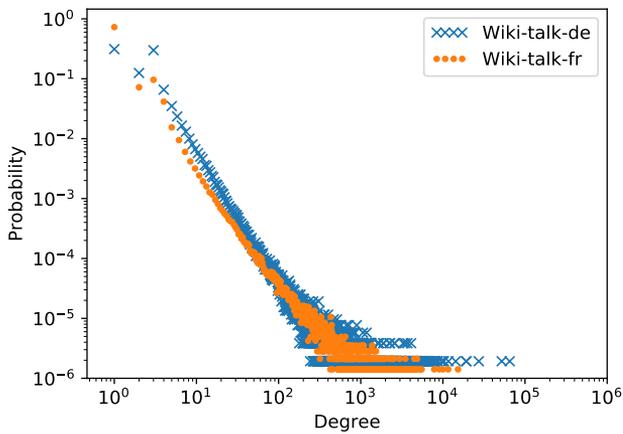
as the target distribution of transformation for the ease of calculation. Combining Equations (4.4), (4.5) and (4.6), we get:

$$x' = \left(\frac{x}{x_{\min}} \right)^{\alpha-1}, \quad (4.7)$$

where x_{\min} is the minimum value of x , and x' is the transformed feature value.



(a) Degree distribution (original)



(b) Degree distribution (transformed)

Figure 4.3: Degree distributions of networks Wiki-talk-de and Wiki-talk-fr, before and after the power-law transformation. Each dot in the plot represents the probability (Y axis) of a degree value (X axis) in the network. Two separated curves overlap after the transformation.

Considering our scenario, degree k or coreness c^{core} starts from 1 in most cases, the formula can thus be further simplified to, for instance,

$$x' = k^{\alpha-1}, \quad k \geq 1. \quad (4.8)$$

In our implementation, we use the power-law transformation for the base features degree, indegree, outdegree, and coreness. We use the method by Clauset et al. [18] to fit a power-law distribution and estimate the exponent α . For features that do not start from 1 such as the indegree and the outdegree, we only perform fitting and transformation on the values that fulfil power-law distribution well.

EXAMPLE Figure 4.3 shows the degree distributions of two networks before and after our transformation. The original curves of degree distributions are clearly separated (as in Figure 4.3a), while being overlapping after the transformation (as in Figure 4.3b). This indicates that our transformation method can transform degree distributions from different networks into a common power-law distribution.

4.2.3.3 PageRank transformation

The standard PageRank of nodes in a network is defined as the stationary probability distribution in a converged random surfing process, *i.e.*, the probability that a surfer is located at a certain node [59]. In a directed graph $G(V, E)$, the PageRank $pr(v)$ of a node v is defined as:

$$pr_v = (1 - \alpha) \sum_{(u,v) \in E} \frac{pr_u}{k_u^{out}} + \frac{\alpha}{|V|}, \quad (4.9)$$

where α is the random teleportation parameter, and k_u^{out} is the outdegree of u .

PageRank is usually used to measure the centrality of nodes in directed networks. However, PageRank is not applicable to compare nodes from different networks, because it is not independent of the network size. Berberich et al. have pointed out that, as a network gets larger, the PageRank values of nodes tend to get smaller. In order to overcome this problem, they have proposed *normalised PageRank* [8], which is defined by:

$$\hat{pr}_v = \frac{pr_v}{pr_{low}}, \quad (4.10)$$

See Section 2.5.4 for details about PageRank.

where pr_{low} is the theoretical lower bound of the PageRank considering the random teleportation at each step of the random walk and at dangling nodes (i.e., nodes with outdegree zero):

$$pr_{\text{low}} = \frac{1}{|V|} \left(\alpha + (1 - \alpha) \sum_{w \in D} pr_w \right), \quad (4.11)$$

where $D \subseteq V$ denotes the set of dangling nodes in the network. The normalised PageRank has been proved to be independent of network size and comparable across networks [8]. Hence, we use it as a transformation for our base feature PageRank.

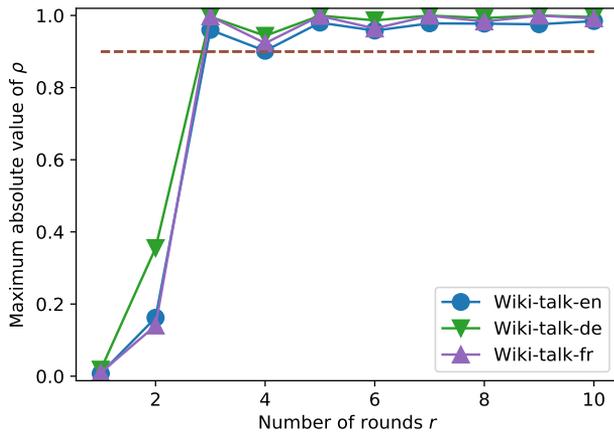
4.2.4 Feature aggregation

It is important to notice that, inside a network, one can characterise a node not only by who it is, but also who are its neighbours, and where it is located. In terms of machine learning, we do not only consider a node's local features, but also look into its neighbourhood's features and the network structure around it. Inspired by the idea of *recursive features* proposed in [32], for each node in the network, we generate its *neighbourhood features* by aggregating its neighbours' features step by step. For more details, in the first round, for each node and each local feature, we compute the average feature value of its neighbours, and store it as a new feature. In the following rounds, we aggregate the features that we get in the last round in the same way.

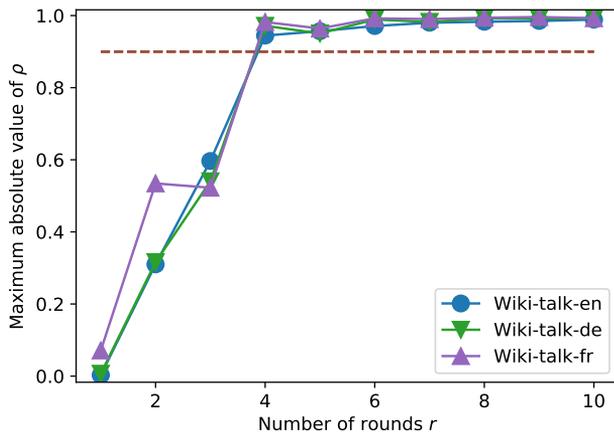
Obviously, this repetitive progress can be done infinitely without the limitation of a round number. And if we continue this repetitive progress, the resulting feature will converge to a certain vector which is the right eigenvector of the \mathbf{P} matrix [51], thus will provide less and less information and will increase the computational overhead. Hence, we introduce a parameter r to limit the number of rounds that we perform feature aggregation.

See Section 2.1.3 for details about eigenvectors. See Section 2.5.3 about the \mathbf{P} matrix.

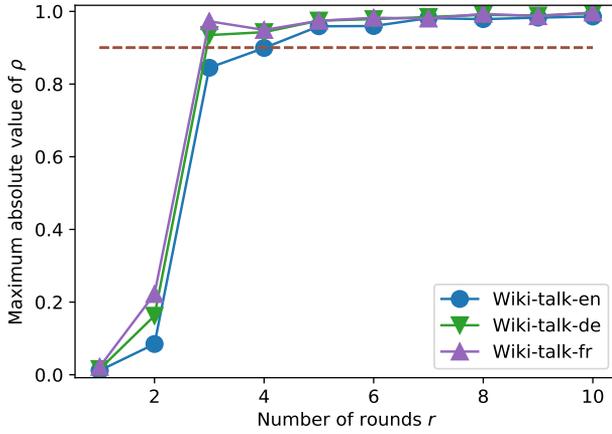
We now explain the vectorised version of the feature aggregation algorithm. With the adjacency matrix \mathbf{A} of the network, and the base feature matrix \mathbf{X}_0 where $X_{0(u,j)}$ is the j th base feature value of node u after feature transformation, we have a vectorised way to perform the feature aggregation using matrix multiplication.



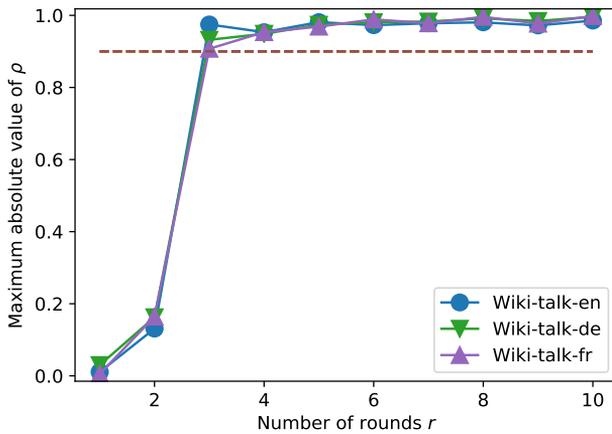
(a) Degree



(b) Local clustering coefficient



(c) PageRank



(d) Coreness

Figure 4.4: Convergence procedure of aggregating neighbourhood features for (a) degree, (b) local clustering coefficient, (c) PageRank and (d) coreness in different datasets (each curve represents a dataset). The X axis shows the round number of feature aggregation, while the Y axis shows the maximum absolute value of the Pearson correlation coefficient ρ between the newly generated neighbourhood features of the current round and the previous round. New features with bigger ρ provide less information [29]. In most cases, ρ gets larger than 0.9 (indicated by the dashed horizontal lines) within 5 rounds. We omit other datasets and features here, since they show similar patterns.

4.2.5 Reduction to classification problem

In the training part of our transfer learning procedure, once we get the feature matrix \mathbf{X}_S for the source dataset, with the labelled user roles \mathbf{y}_S , the problem of user role prediction reduces to a classification problem. Hence, we are able to train a classifier that classifies nodes (users) into their correct classes (labels).

In the training part, we optimise a predictive function $y_{\text{predict}} = f(\mathbf{x})$ that maps the input data (*i.e.*, a node's features \mathbf{x}) into a user label $y_{\text{predict}} \in \{1, 2, \dots, l\}$ with a certain probability, such that $\mathbf{y}_{\text{predict}}$ best matches \mathbf{y}_S [11, 30].

Such a predictive function f can be regarded as a user label classifier which classifies nodes into different labels in other networks, since the features in \mathbf{X}_S are expected to be non-domain-specific, and are already transformed in a way that they can match across networks. In the application part, we use f to compute $\mathbf{y}_{\text{predict}}$ for all nodes in G_T in order to predict user labels in the target network.

The actual classification and optimisation algorithm is however not the main focus of the dissertation. In our implementation, we use a random forest classifier [45]. The parameters of the random forest classifier are set according to the method used by Oshiro and colleagues [58]. Once the classifier is trained, it is able to return the probability that each node (user) belongs to each class (label), given the structural features of the node as input. And it can be saved to classify user labels in multiple target datasets.

In the application part, once we obtain the feature matrix \mathbf{X}_T for the target dataset, we can use the pre-trained classifier to predict the user labels \mathbf{y}_T in the target network.

4.3 EVALUATION

In this section, we show the experimental evaluation of the role transfer task in the Wikipedia user interaction networks. We also show the task of identifying trusted users in ARIS Community as another use case of TraNet. We compare the performance of TraNet with other approaches in terms of the ROC-AUC scores.

4.3.1 Evaluation metric

In the experiments, we use the ROC-AUC score to measure the performance of the classifiers. ROC stands for “receiver operating characteristic curve”. Classification results typically come with the form of probabilities which allows us to choose different thresholds of probability for the classification, and adjust the true positive rate and the false positive rate accordingly. The ROC displays a graph of the true positive rate against the false positive rate of the classifier [26]. The ROC-AUC score measures the area under the ROC, thus it considers both the true positive rate and the false positive rate, under different probability thresholds. Higher ROC-AUC score indicates better performance of the classifier.

4.3.2 Baselines

We choose the following baselines to compare the performance of our approach, TraNet. We have also tried approaches such as the transfer component analysis (TCA) [60], but have found that they do not scale to suit our applications.

- None: training a model from the source network and directly applying it to the target network. This serves as a lower-bound baseline, since no feature transformation is done.
- Trad.: traditional machine learning (*i.e.*, training a model from partial data in a network and apply it to the rest data in the same network). This serves as an upper-bound baseline, since training and test data are sampled from the same domain, and no transformation is necessary.
- SVD: performing feature transformation based on the singular value decomposition (SVD) proposed by Agirre and De La calle [2].
- TrAdaBoost: performing transfer learning with TrAdaBoost proposed by Dai et al. [19]. TrAdaBoost has a different setting from ours: It requires partially labelled data from the target network.

4.3.3 Evaluation of role transfer in Wiki-talk networks

In this subsection, we use Wiki-talk to evaluate the performance of our transfer learning approach, and try to identify administrators among bots and normal users.

4.3.3.1 Administrator classifier

We use all 14 sub-datasets in Wiki-talk with sufficiently many (≥ 25) users labelled as “Administrators”, and build 14 binary classifiers for administrators respectively. Each of the classifier is then applied to predict the administrators in the other 13 sub-datasets. Therefore, we evaluate the classification performance of 182 pairs of source and target datasets.

As shown in Figure 4.5a, we can achieve high ROC-AUC when we use traditional machine learning, which means identifying administrators in the network is achievable given the network structure using our approach. Transfer learning with SVD and TrAdaBoost is ineffective with decreased performance (0.894 and 0.590 on average, respectively), compared with the transfer learning without feature transformation (0.971 on average). Our transfer learning approach can achieve the best performance with an average ROC-AUC of 0.982, improving by more than 1% compared with the transfer learning without feature transformation (0.971 on average).

4.3.3.2 Bot classifier

Similarly, we train 20 bot classifiers respectively from the Wiki-talk networks with sufficiently many (≥ 50) users labelled as “Bots”. Each of the classifier is then applied to predict the bots in the other 19 networks. Therefore, we evaluate the classification performance of 380 pairs of source and target datasets.

The result is shown in Figure 4.5b. The ROC-AUC is only 0.713 for traditional machine learning. This means identifying bots is a relatively more challenging task compared with identifying administrators. Besides, similar to the situation in the administrator classifier, transfer learning with SVD and TrAdaBoost shows decreased performance (0.658 and 0.538 on average, respectively), compared with the transfer learning without feature transformation (0.666 on average). Our transfer learning approach TraNet can achieve an average ROC-AUC

See Table 2 in Appendix C for meta information of the Wiki-talk networks.

of 0.683, with an improvement of 2.5% compared with the transfer learning without feature transformation (0.666 on average).

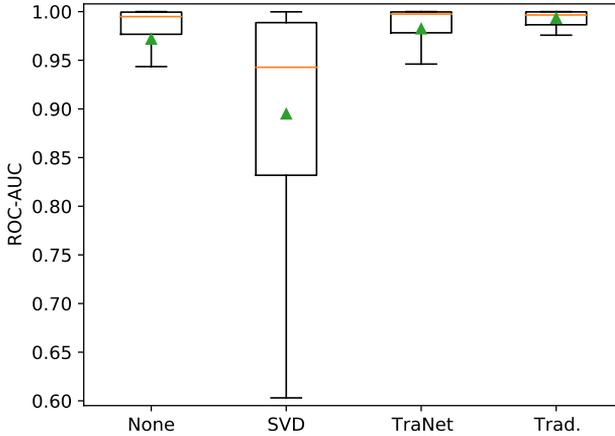
4.3.4 Evaluation of trust transfer in signed networks

In the previous subsection, we showed how our approach, TraNet, can be applied to role analysis tasks in the Wiki-talk datasets. In order to show that our approach can be generalised and applied to other types of transfer learning tasks and heterogeneous types of data, in this subsection, as one more concrete application, we try to apply TraNet to predict trusted users on Web platforms.

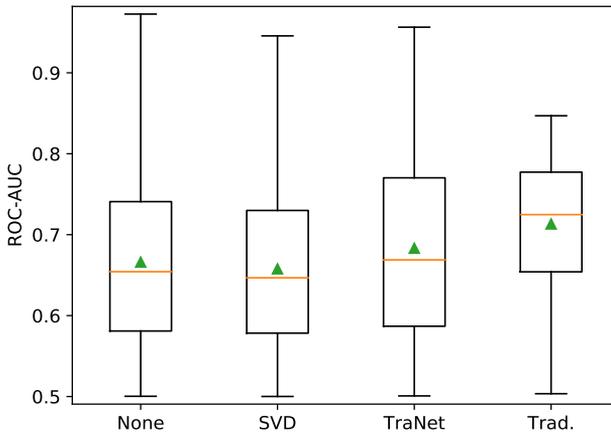
4.3.4.1 Datasets

We now explain the three datasets we use in this experiment. The detailed feature distribution of nodes in these networks can be found in Appendix B.

- ARIS. This dataset contains a user interaction network in Software AG's ARIS Community. The ARIS Community is the Business Process Management (BPM) system used by the Software AG company, to enable better collaboration, engagement and sharing. At the time we extracted it, it had 9,566 threads and 20,538 comments by 4,216 users, and the total user number was 394,716. Each directed edge (User_ID_A, User_ID_B, timestamp) represents a user's comment to another user's post or comment.
- Slashdot-Zoo: This dataset is a signed network extracted from Slashdot, consisting of 79,120 users and 515,397 directed relations [41]. In this network, each directed signed edge represents a "friend" (positive) or "foe" (negative) relation from one user to another on the technology news site Slashdot, where each user can explicitly mark other users as their friends or foes in order to increase or decrease the chance to see their posts. 23.91% edges are negative.
- Epinion-Trust: This dataset is a signed network of Epinions, an online product rating site [48]. It consists of 131,828 users and 841,372 directed, signed edges, each representing a trust (positive) or distrust (negative) relation from one user to any user (possibly herself). 14.70% edges are negative.



(a) Administrator classifier



(b) Bot classifier

Figure 4.5: ROC-AUC performance of the classifiers using different settings (see Section 4.3.2). In each box plot, one data point corresponds to the ROC-AUC score of a classifier that transfers user roles from one Wiki-talk network to another. The orange bar shows the median value, while the green triangle shows the mean value of the ROC-AUC in each experiment. Our approach TraNet achieves the best average ROC-AUC among all transfer learning approaches in both classification tasks. Note TrAdaBoost is omitted here due to its poor performance.

Table 1: Predicting trusted users in the target network ARIS with the knowledge transferred from the two source networks Slashdot-Zoo and Epinion-Trust respectively. The values in the table show the ROC-AUC performance of the classifier in different settings (see Section 4.3.2). We can achieve the best performance with transfer learning using our approach TraNet.

	Source dataset	
	Slashdot-Zoo	Epinion-Trust
None	0.7317	0.7662
SVD	0.6294	0.7203
TrAdaBoost	0.6896	0.7035
TraNet	0.8172	0.7953
Trad.	0.8556	

4.3.4.2 Results

We compute the trusted users in Slashdot-Zoo and Epinion-Trust using the EigenTrust algorithm [36], and use each of them as the source dataset to learn a model respectively, and predict the trusted users in ARIS. The result is shown in Table 1. It shows that performing no feature transformation does not work well. Our approach TraNet outperforms other transfer learning approaches, and can even achieve the performance close to traditional within-network learning when using Slashdot-Zoo as the source dataset.

4.4 SUMMARY

In this chapter, we propose TraNet, a transfer learning approach for predicting user labels in unlabelled networks.

TraNet relies on the measurement of users' social behaviour as node structural features, the transfer of node structural features using feature transformation, the feature aggregation, and the classification of nodes. TraNet provides a novel method to better analyse users and their behaviour on, especially new, online platforms. We conduct experiments on real network datasets with the tasks of user role and trust transfer, and the results show the effectiveness of TraNet.

CONCLUSION

This dissertation has studied two topics in network science: Phenomena in growing networks, and learning across networks.

5.1 CONCLUSION AND OUTLOOK: PHENOMENA IN GROWING NETWORKS

We have systematically reviewed various existing preferential attachment network models, and analysed how they each explains various phenomena in real complex networks. We have reported the empirical findings of the exponential network growth, and the time-invariance of the degree growth in two real citation networks. We have then proposed a novel analytical framework for preferential attachment network models based on the empirical findings. With the new framework we have achieved the following results analytically:

- We have shown that the linear and exponential network growths are the only two cases that are consistent with the time translation symmetric degree growth.
- We have unified different condensation (winner-takes-all) phenomena in complex networks.
- We have proved that ageing can prevent condensation (winner-takes-all) with mild condition.

Our analytical results have been verified with extensive simulations.

For future research, the exponential growth of the network size cannot sustain forever due to the limited number of potential nodes [67], so it has to eventually slow down. Such a slowdown can be realised by relaxing the model assumptions by, for instance, changing the fitness distribution with time whilst still maintaining the time-invariant degree growth. Another possibility is to relax the time-invariance of the degree growth by allowing the parameters in our formalism to vary.

The analytic form of the resulting network growth and its relation to the degree distribution are also interesting to study.

We have based our observations on citation networks. The studied model thus limits itself to no edge removal and edge creation only at node arrival. Besides, its preferential attachment process only considers local information of nodes, such as degree, fitness and ageing. Lifting one or more of these limitations would enable extensions of the model to work with more general networks.

5.2 CONCLUSION AND OUTLOOK: LEARNING ACROSS NETWORKS

We have systematically studied the problem of predicting user labels in unlabelled social networks, and proposed to use transfer learning to transfer knowledge of user labels from known networks.

The proposed approach TraNet relies on the measurement of users' social behaviour as node structural features, the transfer of node structural features using feature transformation, the feature aggregation, and the classification of nodes. We have proposed a method of transformation for power-law distributions. This method can be used effectively in transfer learning tasks in network analysis on features such as node degrees and coreness. We have applied TraNet on the tasks of transferring user roles across the Wikipedia user interaction networks of different languages, and identifying trusted users in the ARIS Community, and the results have shown the effectiveness of TraNet.

Future work in this topic may include extending TraNet and study other problems in network analysis with transfer learning, such as link prediction and friend recommendation. Moreover, we have been focusing on the study of users in this dissertation. TraNet can also be applied to study other entities on the Web, such as groups or products, since they can also be represented as nodes in the network.

APPENDIX: SUPPLEMENTARY RESULT FOR CHAPTER 3

In this appendix we provide the supplementary experimental results for Chapter 3.

A.1 NETWORK GROWTH CURVES OF SYNTHETIC NETWORKS

A.1.1 *Synthetic networks with heterogeneous fitness*

Figure A.1 shows the network size growth curves of the synthetic networks generated from the preferential attachment model with heterogeneous fitness, without ageing (as in the Bianconi-Barabási model) using our framework:

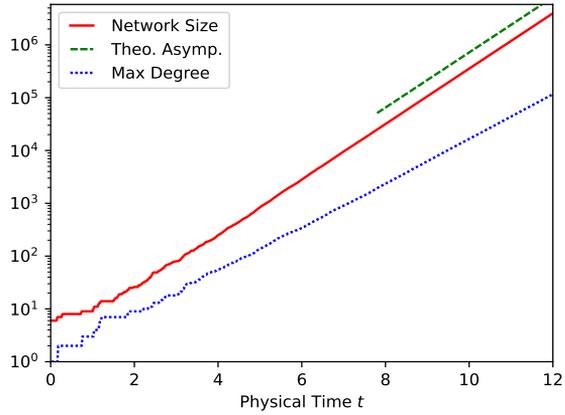
$$\frac{dk_i}{d\tau_i} \sim k_i \eta_i, \quad (\text{A.1})$$

with the fitness distribution

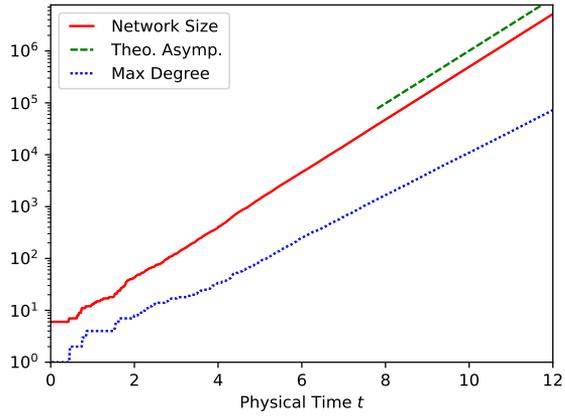
$$\rho(\eta) \sim (1 - \eta)^\lambda, \quad (\text{A.2})$$

where λ is the parameter to be varied. The results for $\lambda = 0.5$ and $\lambda = 1.5$ are already shown in Section 3.6.3.

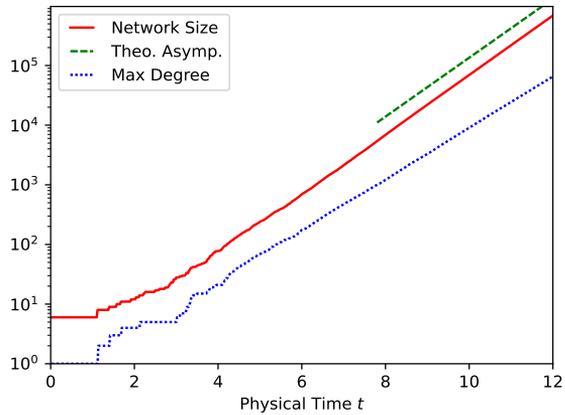
In the plots, the green dashed lines indicate the asymptotic slopes of the theoretical estimation of the network growth. The blue dashed curves show the growth of the maximum degree in the networks. Note that the plots use the linear-log scale.



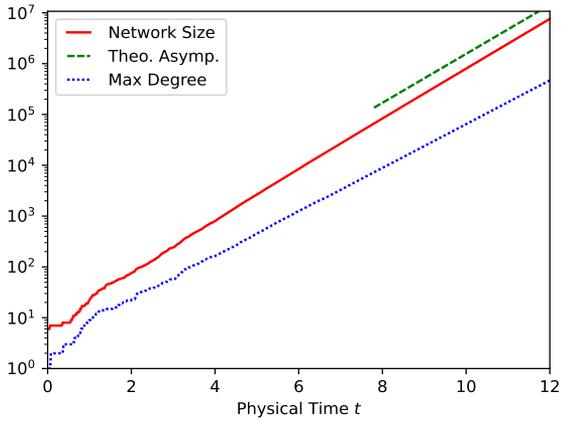
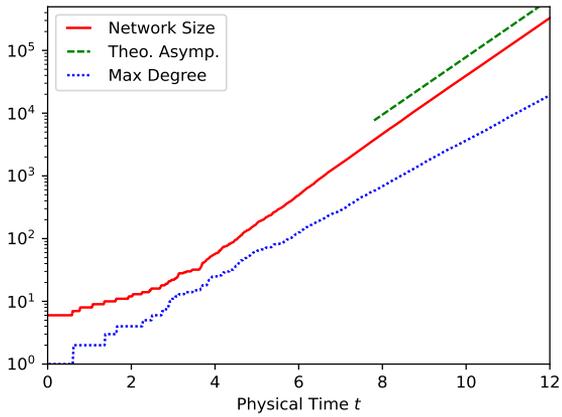
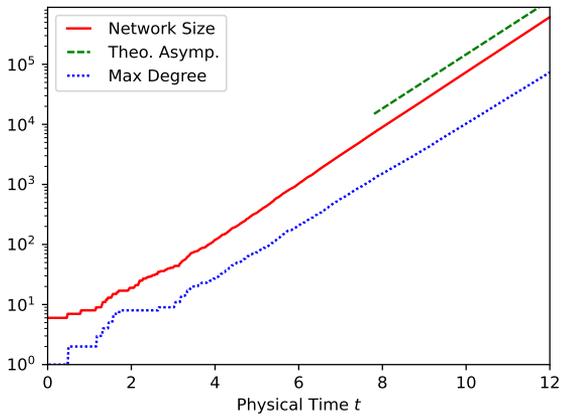
(a) $\lambda = 0.1$

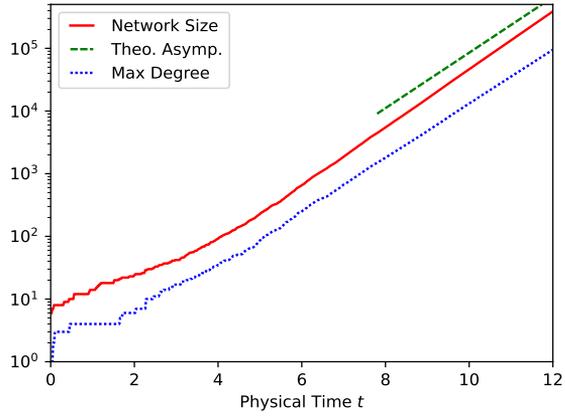


(b) $\lambda = 0.2$

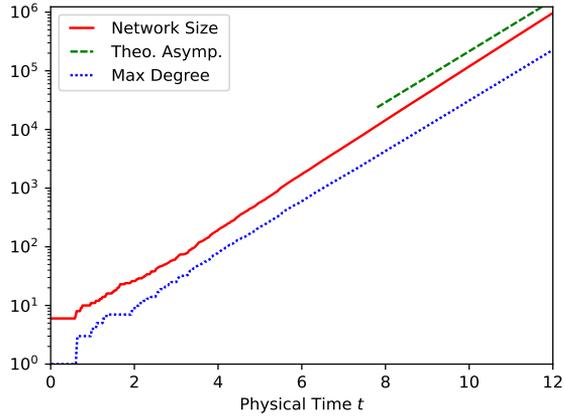


(c) $\lambda = 0.3$

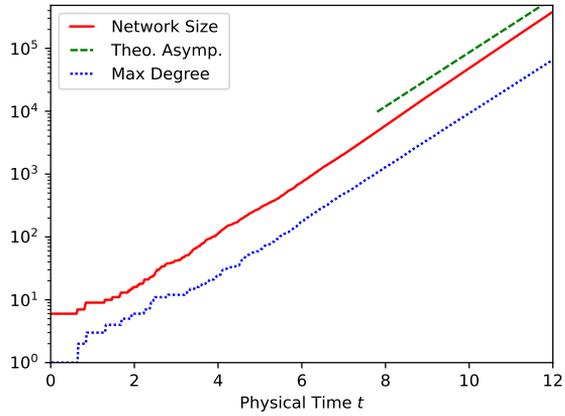
(d) $\lambda = 0.4$ (e) $\lambda = 0.6$ (f) $\lambda = 0.7$



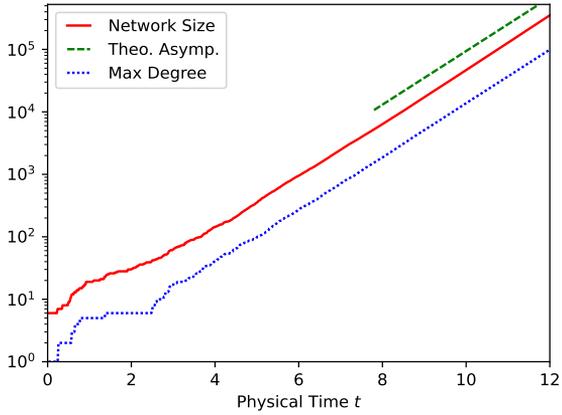
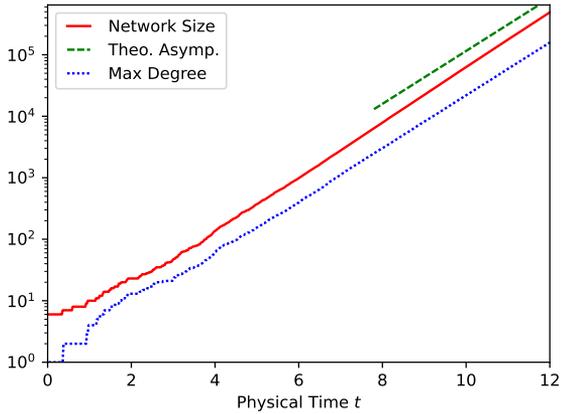
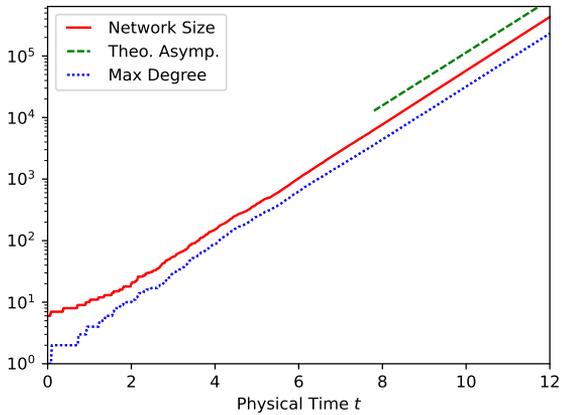
(g) $\lambda = 0.8$

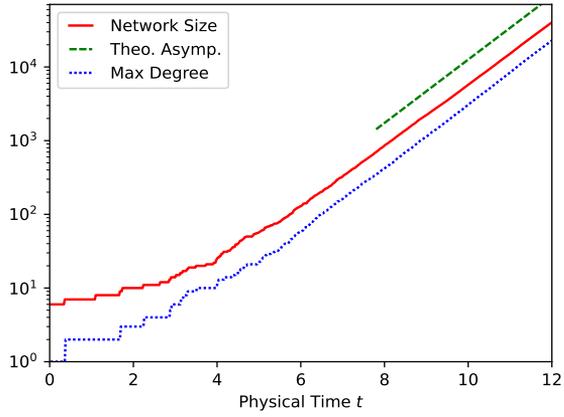


(h) $\lambda = 0.9$

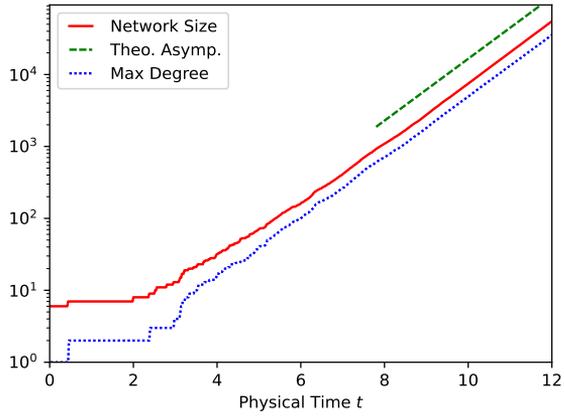


(i) $\lambda = 1.0$

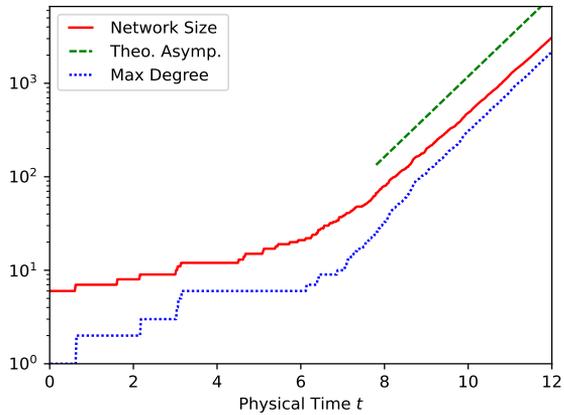
(j) $\lambda = 1.1$ (k) $\lambda = 1.2$ (l) $\lambda = 2.0$



(m) $\lambda = 2.5$



(n) $\lambda = 3.0$



(o) $\lambda = 4.0$

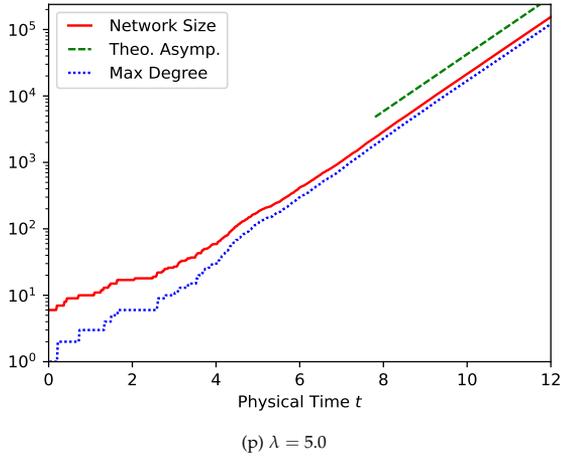


Figure A.1: The network growth curves of synthetic networks generated using our framework with preferential attachment with heterogeneous fitness ($\Pi \sim k\eta$ as in the Bianconi-Barabási model). In each plot the parameter λ of the fitness distribution $\rho(\eta) \sim (1 - \eta)^\lambda$ is set different.

A.1.2 *Synthetic networks with heterogeneous fitness and ageing*

We now show the supplementary experimental results of the network growth curves of synthetic networks generated from the preferential attachment model with heterogeneous fitness and ageing effect (as in the relevance decay model) using our framework:

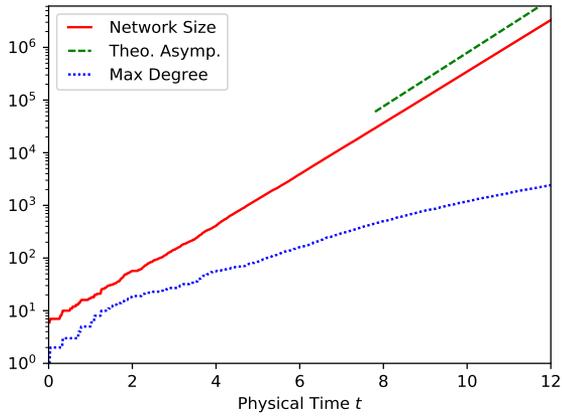
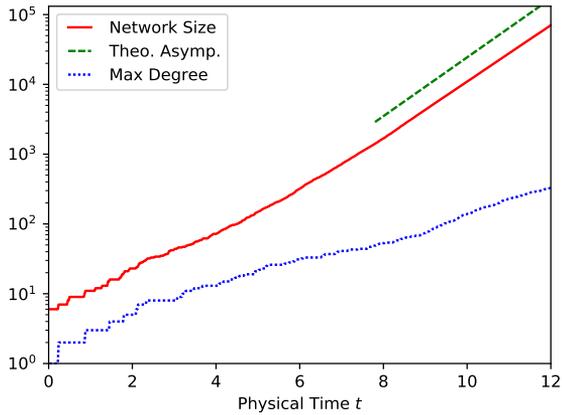
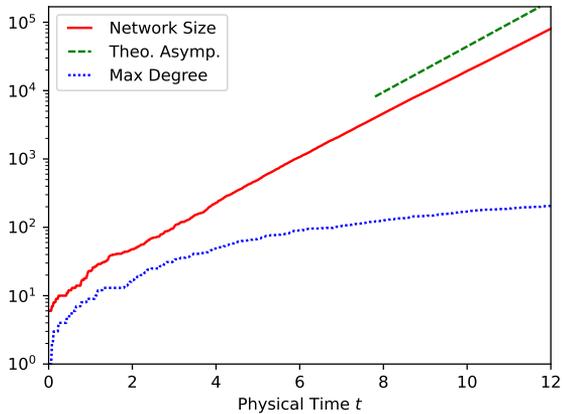
$$\frac{dk_i}{d\tau_i} \sim k_i \eta_i R(\tau), \quad (\text{A.3})$$

with the fitness distribution

$$\rho(\eta) \sim (1 - \eta)^\lambda, \quad (\text{A.4})$$

with different parameters λ and ageing functions $R(\tau)$. We evaluate the power and exponential ageing as discussed in Section 3.3.3.

Figure A.2 shows the network growth curves of synthetic networks with power function ageing. Figure A.3 shows the network growth curves of synthetic networks with exponential ageing. In the plots, the green dashed lines indicate the asymptotic slopes of the theoretical estimation of the network growth. The blue dashed curves show the growth of the maximum degree in the networks. Note that the plots use the linear-log scale.

(a) $\lambda = 2.0, \gamma = 0.8$ (b) $\lambda = 2.0, \gamma = 1.0$ (c) $\lambda = 2.0, \gamma = 1.2$

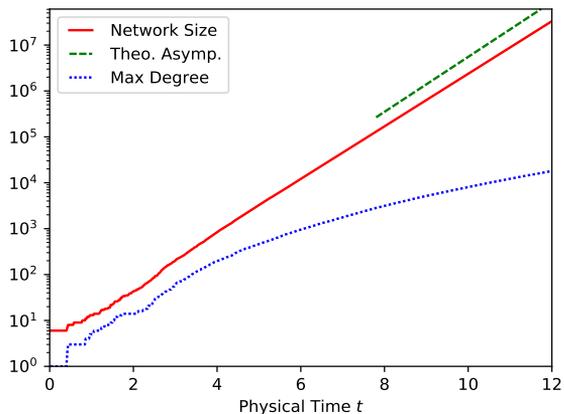
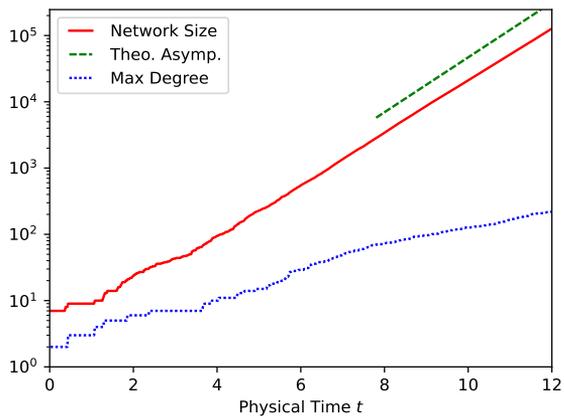
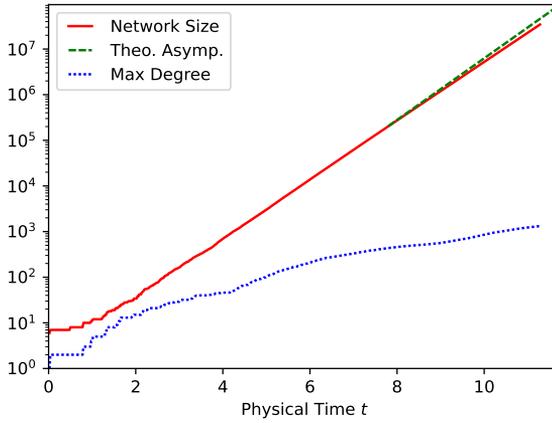
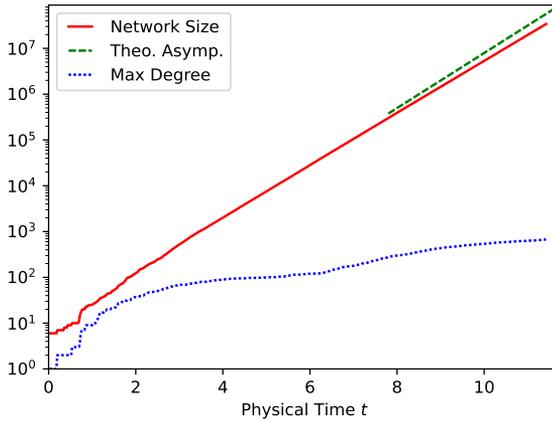
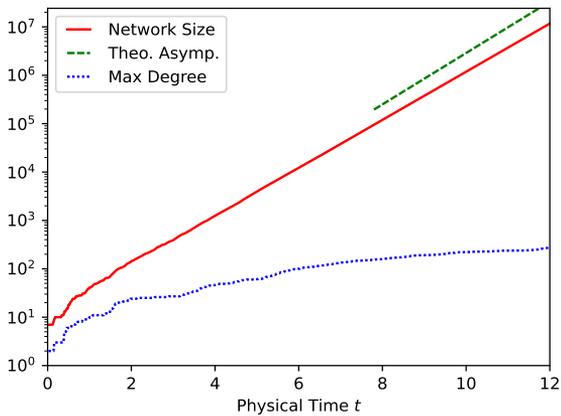
(d) $\lambda = 1.5, \gamma = 0.8$ (e) $\lambda = 1.5, \gamma = 1.2$

Figure A.2: The network growth curves of the synthetic networks generated using our framework with preferential attachment with heterogeneous fitness and power ageing ($\Pi \sim k\eta R(\tau)$). In each plot, the parameters λ of the fitness distribution $\rho(\eta) \sim (1 - \eta)^\lambda$, and γ of the power ageing function $R(\tau) = (\tau + 1)^{-\gamma}$, are set different.

(a) $\lambda = 1.2, \gamma = 0.5$ (b) $\lambda = 1.2, \gamma = 0.6$ (c) $\lambda = 1.2, \gamma = 0.7$

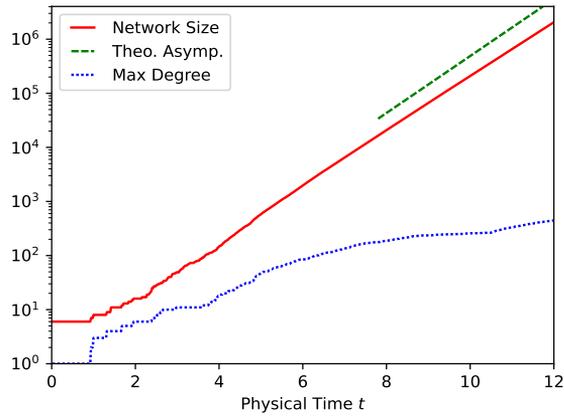
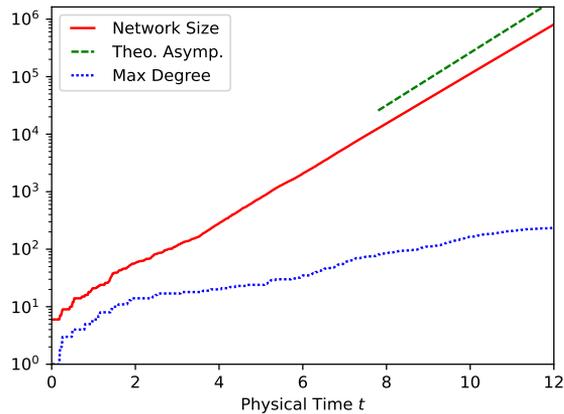
(d) $\lambda = 1.5, \gamma = 0.6$ (e) $\lambda = 1.5, \gamma = 0.7$

Figure A.3: The network growth curves of the synthetic networks generated using our framework with preferential attachment with heterogeneous fitness and exponential ageing ($\Pi \sim k\eta R(\tau)$). In each plot, the parameters λ of the fitness distribution $\rho(\eta) \sim (1 - \eta)^\lambda$, and γ of the exponential ageing function $R(\tau) = e^{-\gamma\tau}$, are set different.

A.2 DEGREE DISTRIBUTIONS OF SYNTHETIC NETWORKS

A.2.1 *Synthetic networks with heterogeneous fitness*

Figure A.4 shows the degree distributions of the synthetic networks generated from the preferential attachment model with heterogeneous fitness, without ageing (as in the Bianconi-Barabási model) using our framework:

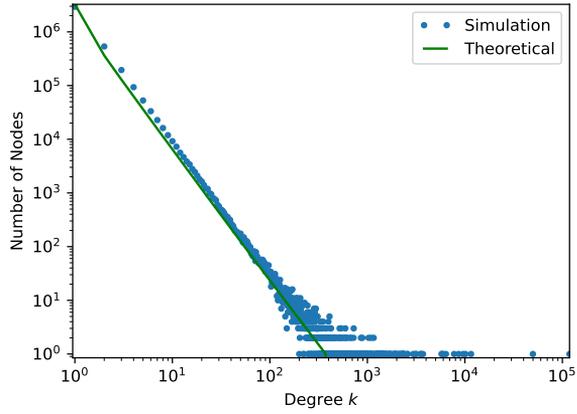
$$\frac{dk_i}{d\tau_i} \sim k_i \eta_i, \quad (\text{A.5})$$

with the fitness distribution

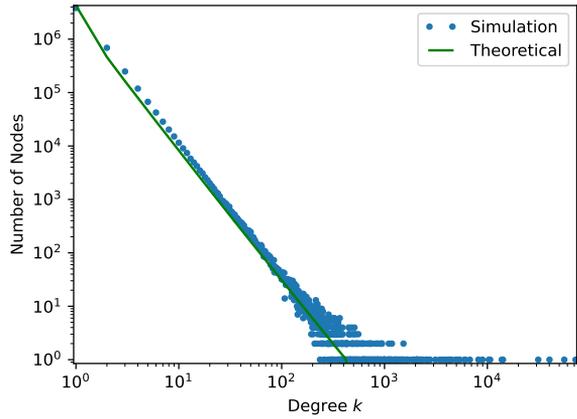
$$\rho(\eta) \sim (1 - \eta)^\lambda, \quad (\text{A.6})$$

where λ is the parameter to be varied. The results for $\lambda = 0.5$ and $\lambda = 1.5$ are already shown in Section 3.6.3.

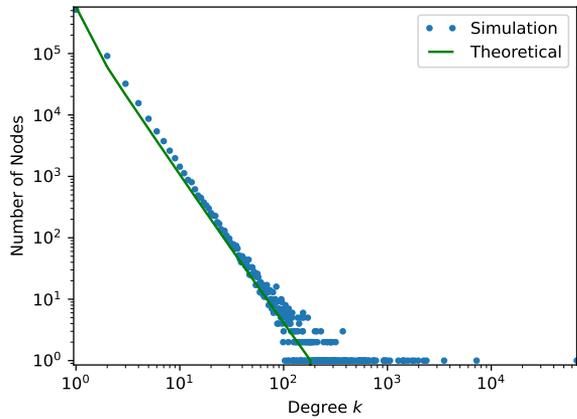
In the plots, the green curves indicate the theoretical estimation of the degree distributions. Note that the plots use the log-log scale.



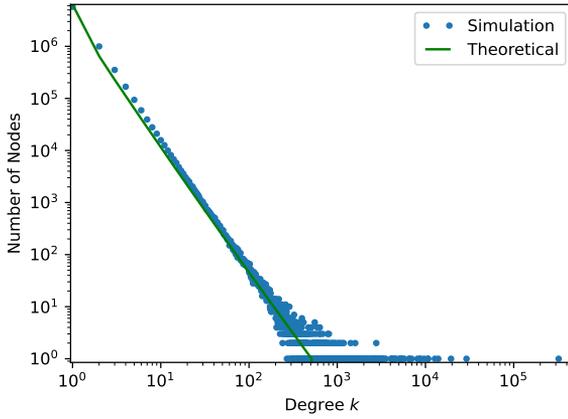
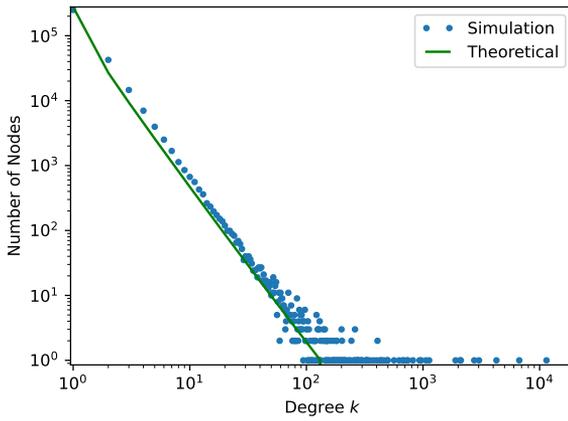
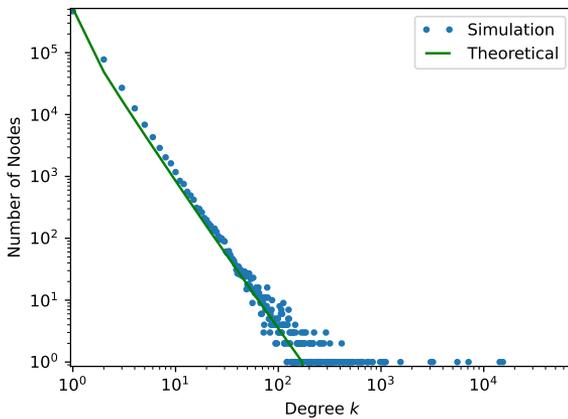
(a) $\lambda = 0.1$

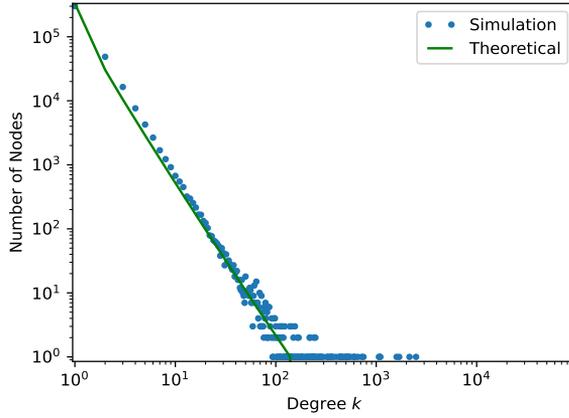


(b) $\lambda = 0.2$

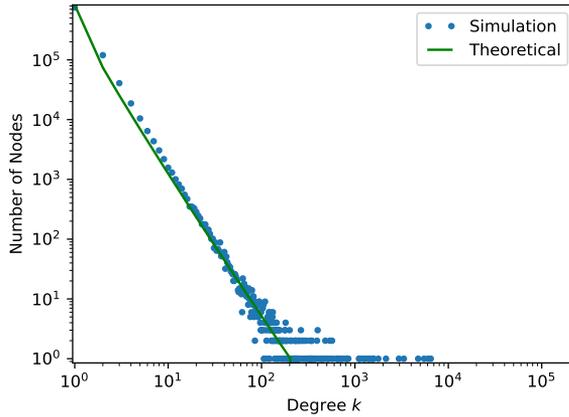


(c) $\lambda = 0.3$

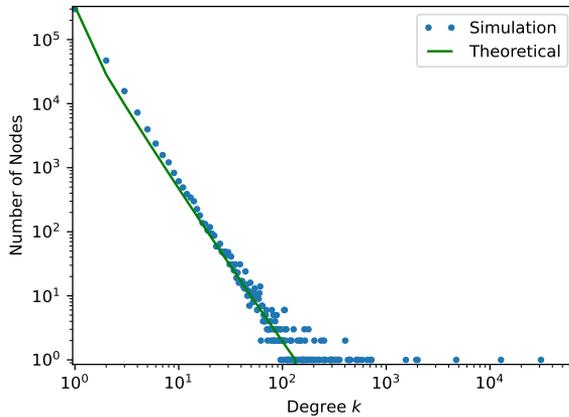
(d) $\lambda = 0.4$ (e) $\lambda = 0.6$ (f) $\lambda = 0.7$



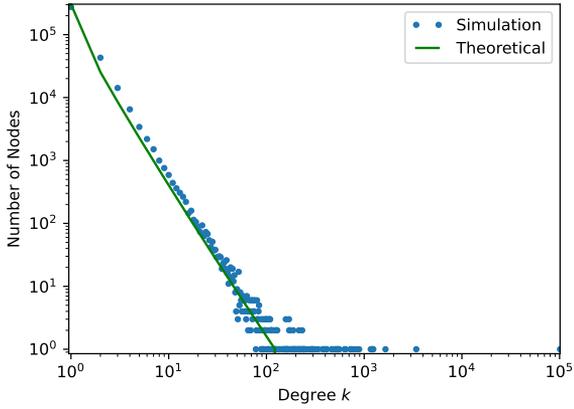
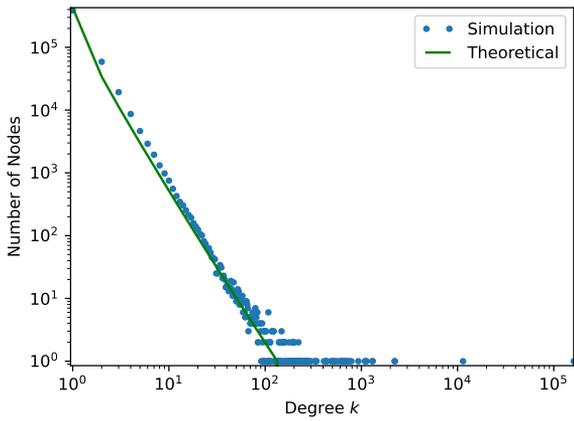
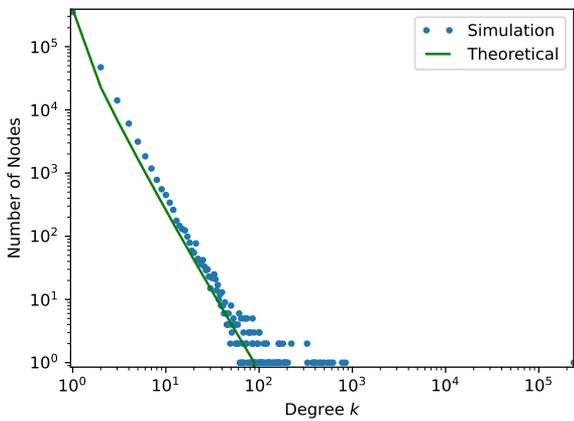
(g) $\lambda = 0.8$

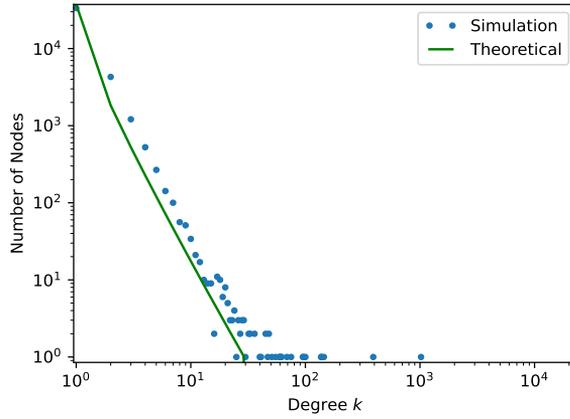


(h) $\lambda = 0.9$

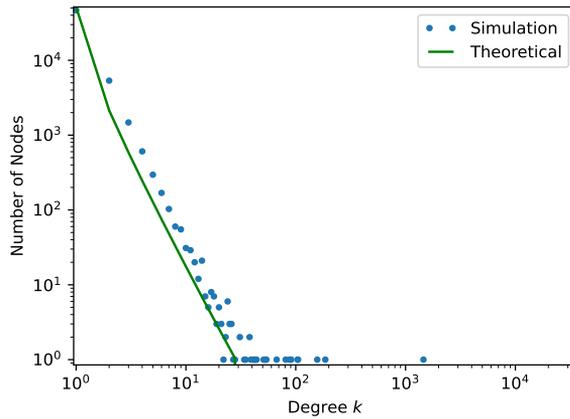


(i) $\lambda = 1.0$

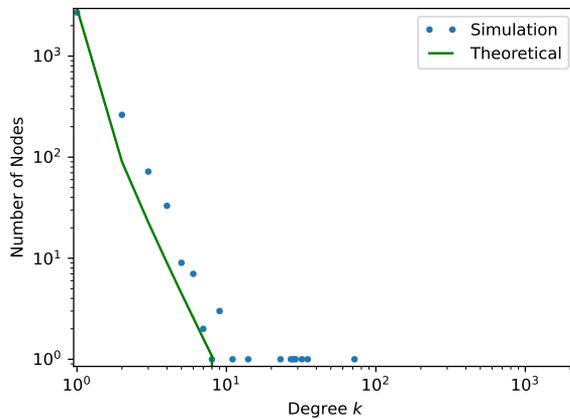
(j) $\lambda = 1.1$ (k) $\lambda = 1.2$ (l) $\lambda = 2.0$



(m) $\lambda = 2.5$



(n) $\lambda = 3.0$



(o) $\lambda = 4.0$

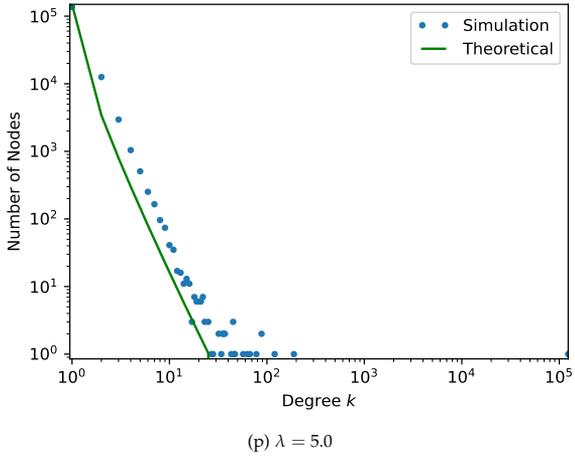


Figure A.4: The degree distributions of the synthetic networks generated using our framework with preferential attachment with heterogeneous fitness ($\Pi \sim k\eta$ as in the Bianconi-Barabási model). In each plot the parameter λ of the fitness distribution $\rho(\eta) \sim (1 - \eta)^\lambda$ is set different.

A.2.2 Synthetic networks with heterogeneous fitness and ageing

We now show the supplementary experimental results of the degree distributions of synthetic networks generated from the preferential attachment model with heterogeneous fitness and ageing effect (as in the relevance decay model) using our framework:

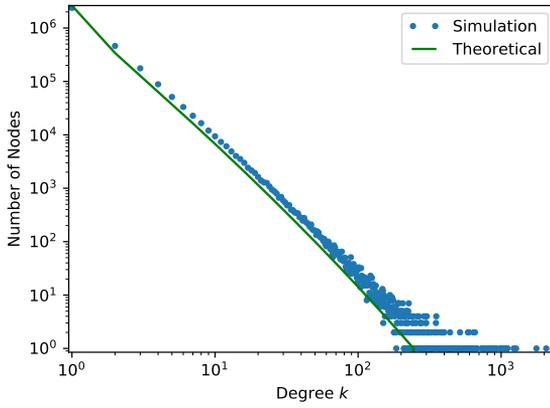
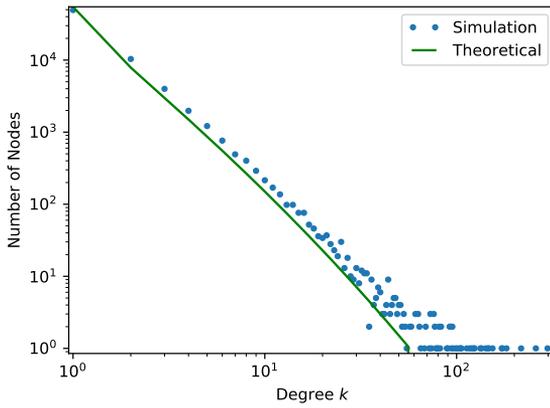
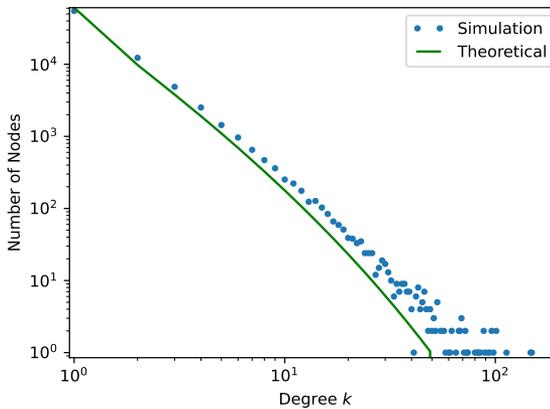
$$\frac{dk_i}{d\tau_i} \sim k_i \eta_i R(\tau), \quad (\text{A.7})$$

with the fitness distribution

$$\rho(\eta) \sim (1 - \eta)^\lambda, \quad (\text{A.8})$$

with different parameters λ and ageing functions $R(\tau)$. We evaluate the power and exponential ageing as discussed in Section 3.3.3.

Figure A.5 shows the degree distributions of synthetic networks with power function ageing. Figure A.6 shows the degree distributions of synthetic networks with exponential ageing. In the plots, the green curves indicate the theoretical estimation of the degree distributions. Note that the plots use the log-log scale.

(a) $\lambda = 2.0, \gamma = 0.8$ (b) $\lambda = 2.0, \gamma = 1.0$ (c) $\lambda = 2.0, \gamma = 1.2$

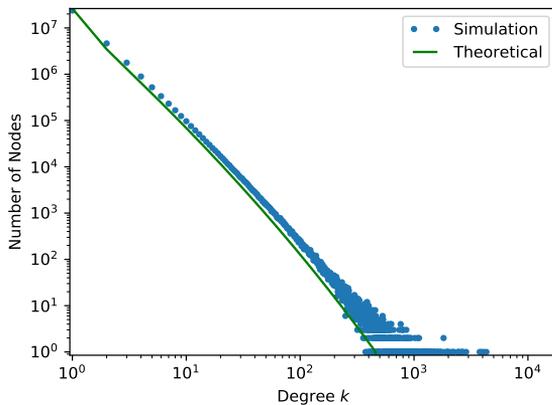
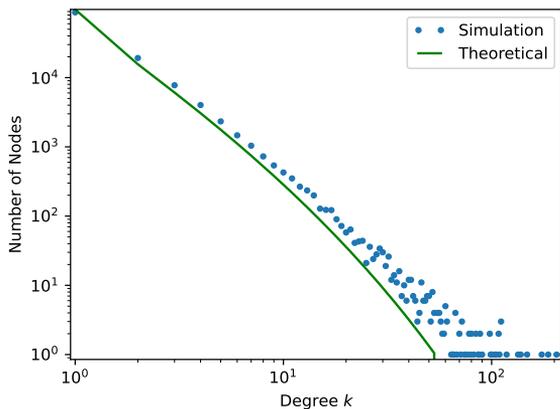
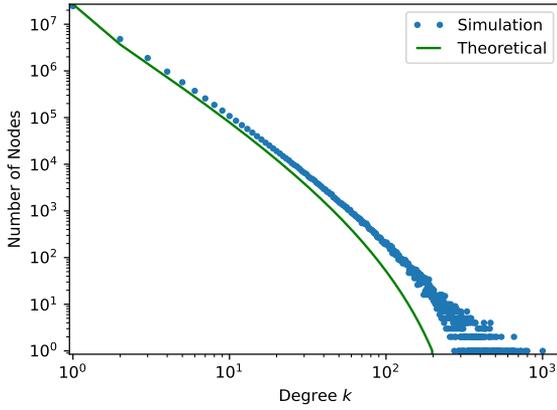
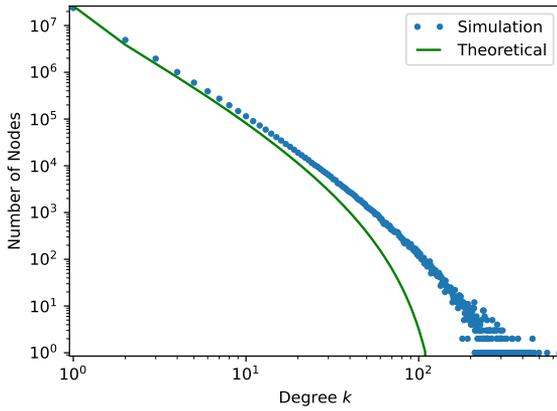
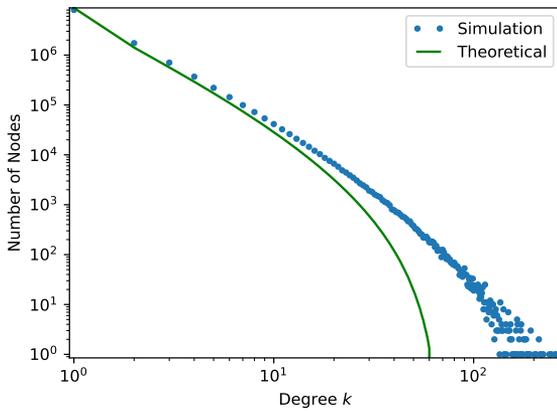
(d) $\lambda = 1.5, \gamma = 0.8$ (e) $\lambda = 1.5, \gamma = 1.2$

Figure A.5: The degree distributions of the synthetic networks generated using our framework with preferential attachment with heterogeneous fitness and power ageing ($\Pi \sim k\eta R(\tau)$). In each plot, the parameters λ of the fitness distribution $\rho(\eta) \sim (1-\eta)^\lambda$, and γ of the power ageing function $R(\tau) = (\tau+1)^{-\gamma}$, are set different.

(a) $\lambda = 1.2, \gamma = 0.5$ (b) $\lambda = 1.2, \gamma = 0.6$ (c) $\lambda = 1.2, \gamma = 0.7$

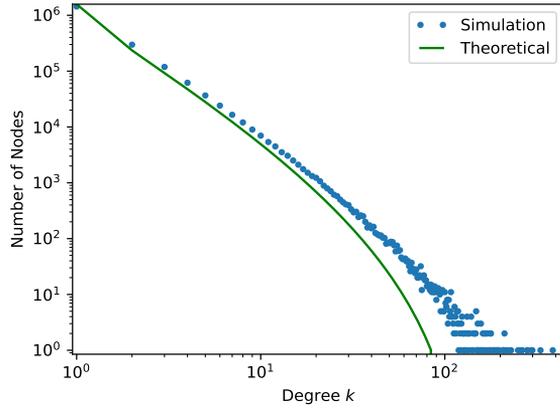
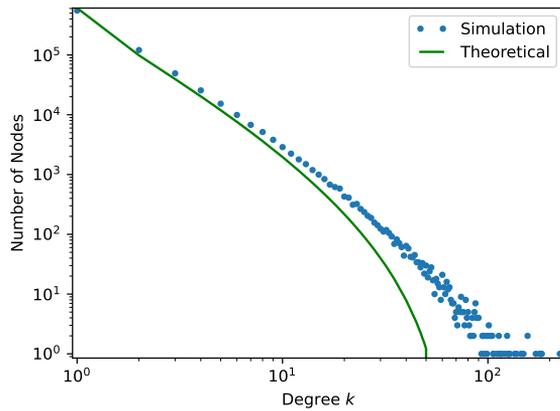
(d) $\lambda = 1.5, \gamma = 0.6$ (e) $\lambda = 1.5, \gamma = 0.7$

Figure A.6: The degree distributions of the synthetic networks generated using our framework with preferential attachment with heterogeneous fitness and exponential ageing ($\Pi \sim k\eta R(\tau)$). In each plot, the parameters λ of the fitness distribution $\rho(\eta) \sim (1-\eta)^\lambda$, and γ of the exponential ageing function $R(\tau) = e^{-\gamma\tau}$, are set different.

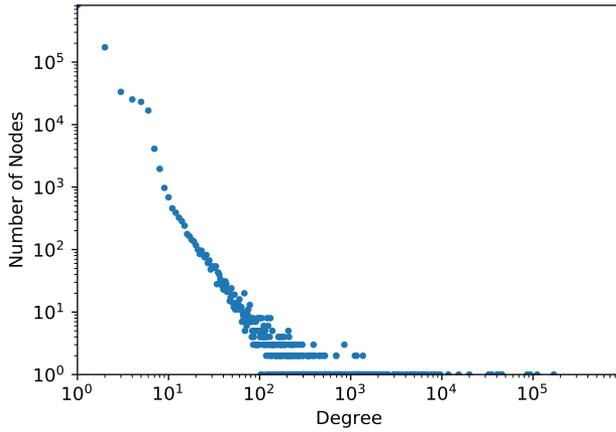
B

APPENDIX: SUPPLEMENTARY RESULT FOR CHAPTER 4

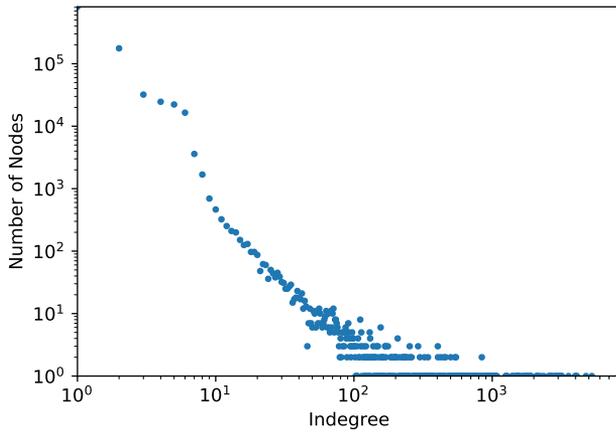
In this appendix we provide the supplementary results for Chapter 4.

B.1 BASE FEATURE DISTRIBUTIONS OF WIKI-TALK NETWORKS

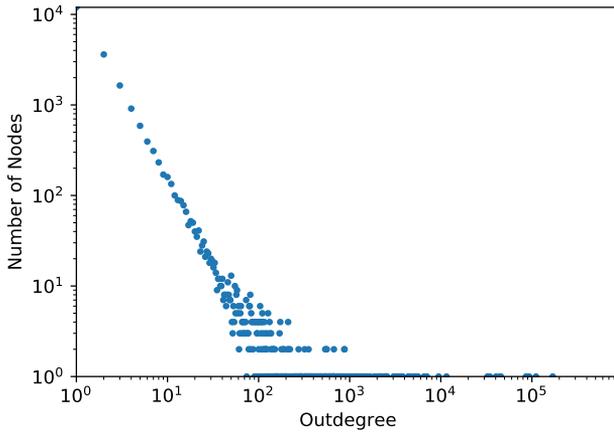
In the following we show the base feature distributions of nodes in the Wiki-talk networks, apart from the ones of Wiki-talk-en which are shown in Figure B.5 in Section 4.2.2.



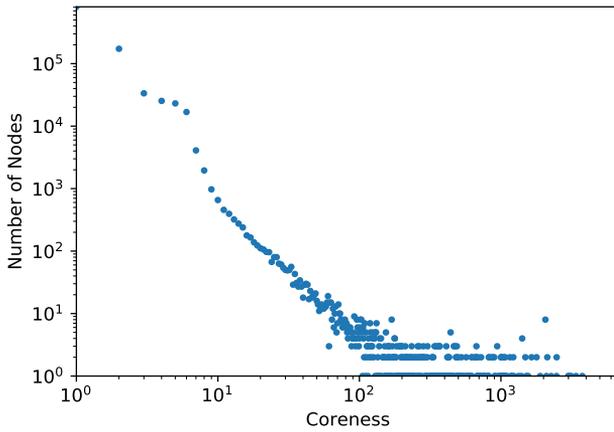
(a) Degree distribution



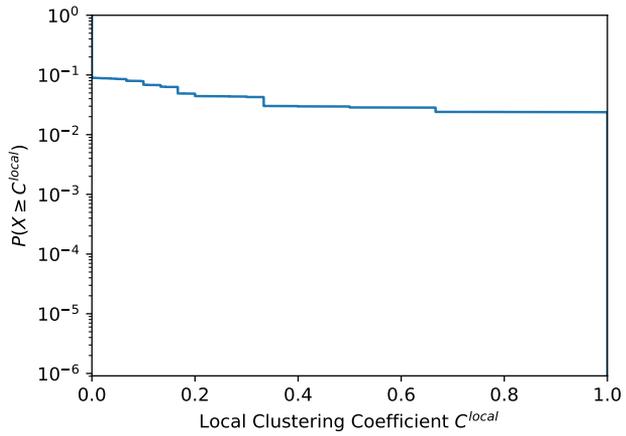
(b) Indegree distribution



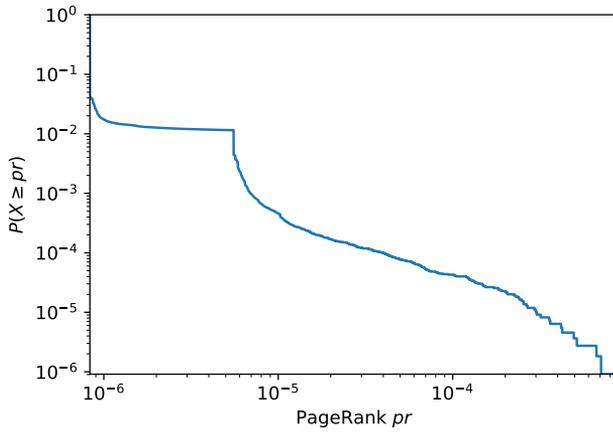
(c) Outdegree distribution



(d) Coreness distribution

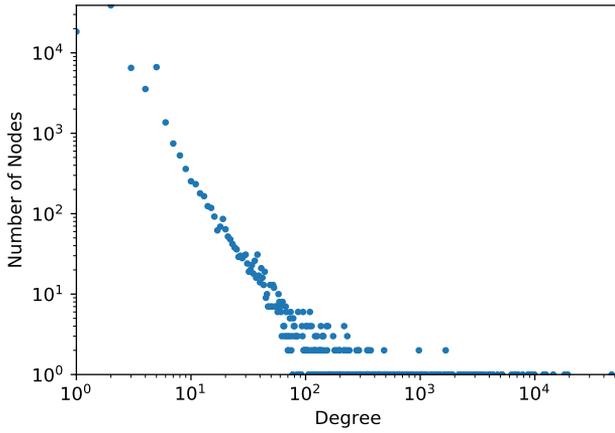


(e) Local clustering coefficient distribution

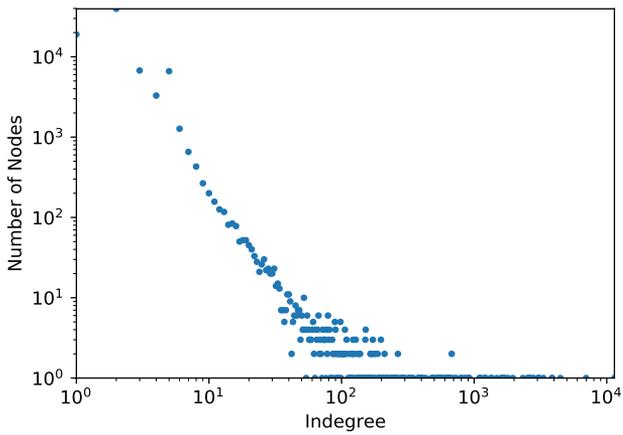


(f) PageRank distribution

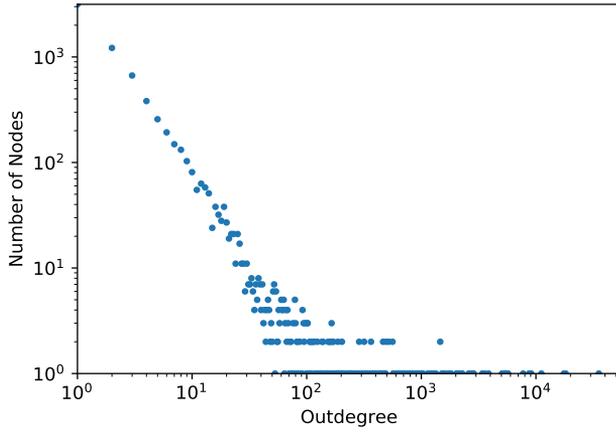
Figure B.1: The base feature distribution of the Wiki-talk-ar network.



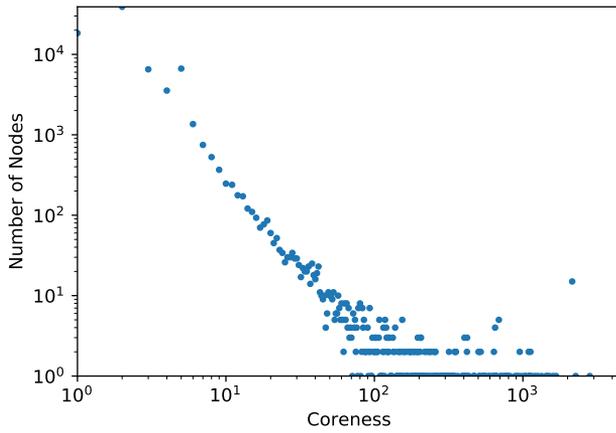
(a) Degree distribution



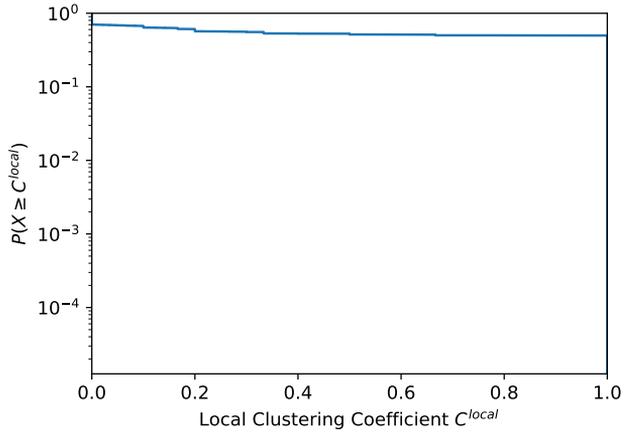
(b) Indegree distribution



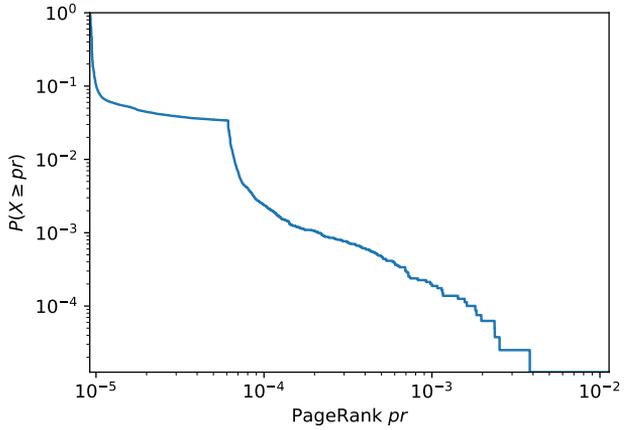
(c) Outdegree distribution



(d) Coreness distribution

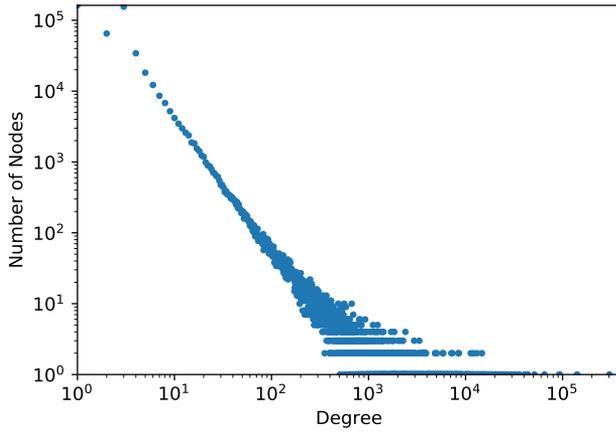


(e) Local clustering coefficient distribution

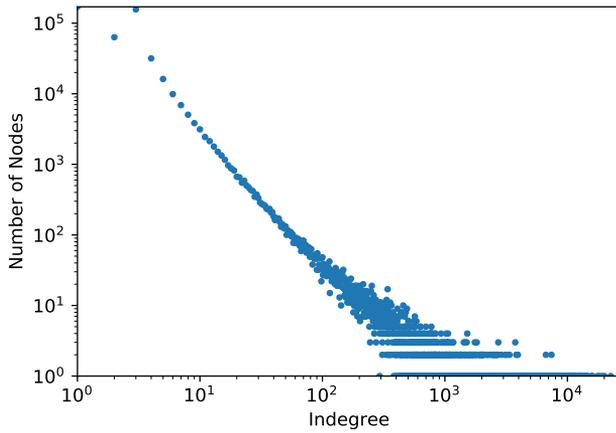


(f) PageRank distribution

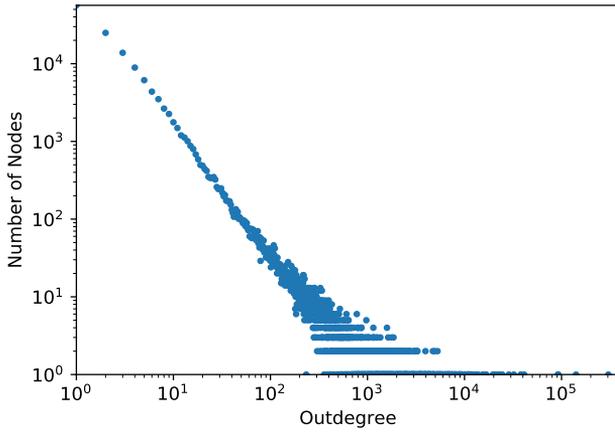
Figure B.2: The base feature distribution of the Wiki-talk-ca network.



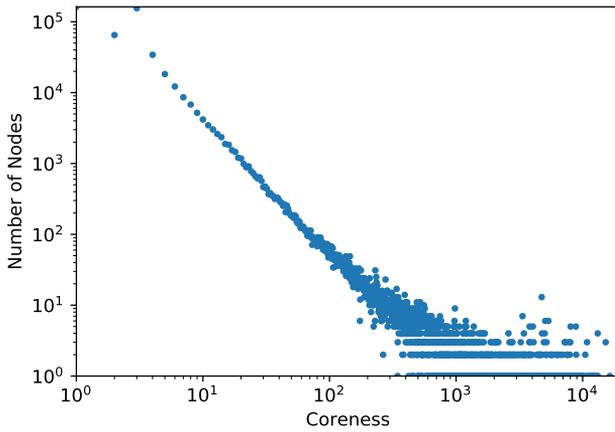
(a) Degree distribution



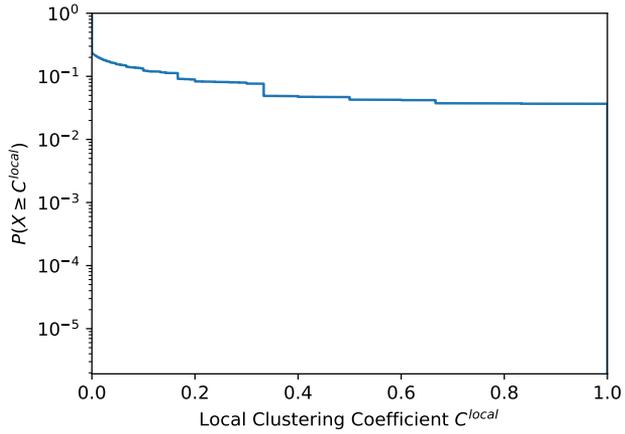
(b) Indegree distribution



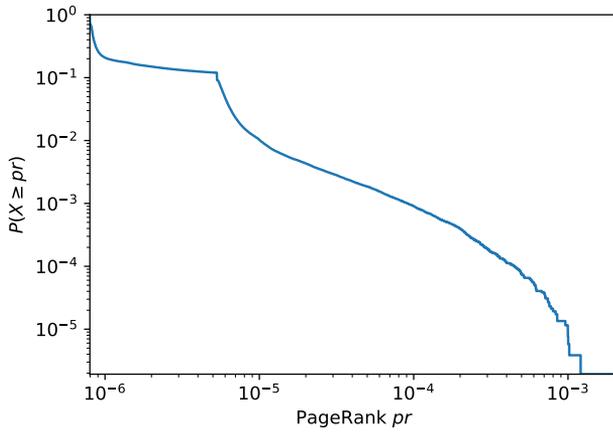
(c) Outdegree distribution



(d) Coreness distribution

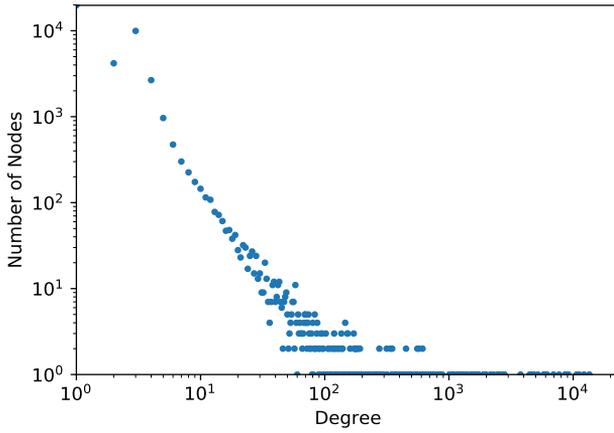


(e) Local clustering coefficient distribution

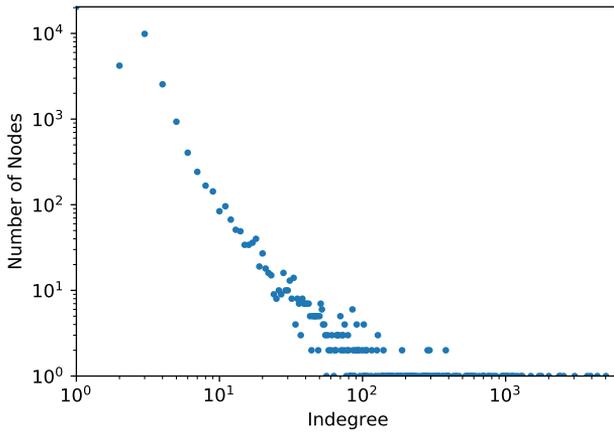


(f) PageRank distribution

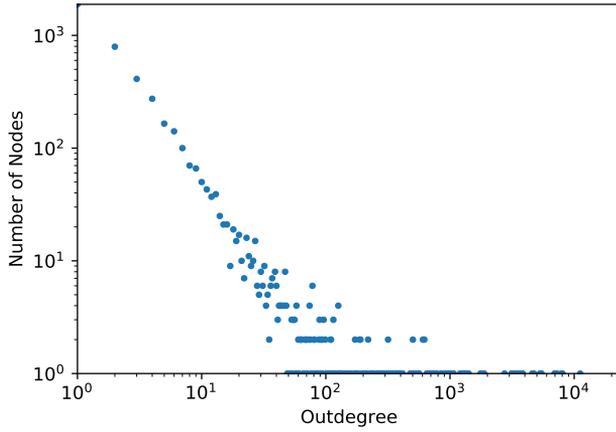
Figure B.3: The base feature distribution of the Wiki-talk-de network.



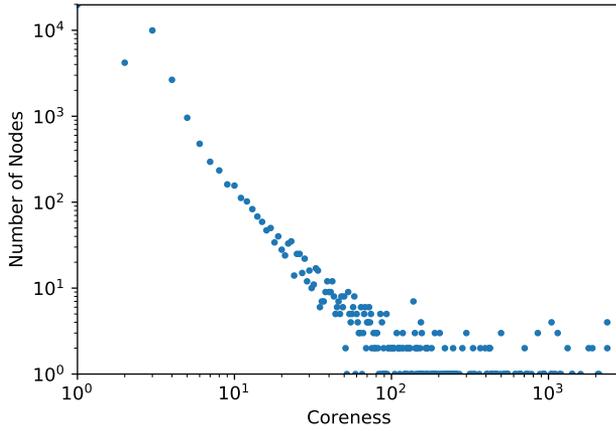
(a) Degree distribution



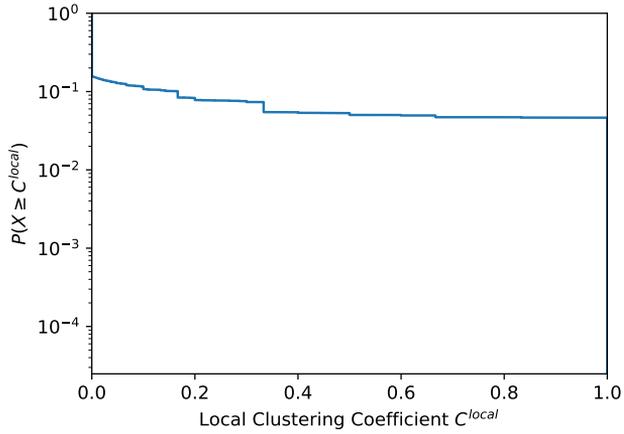
(b) Indegree distribution



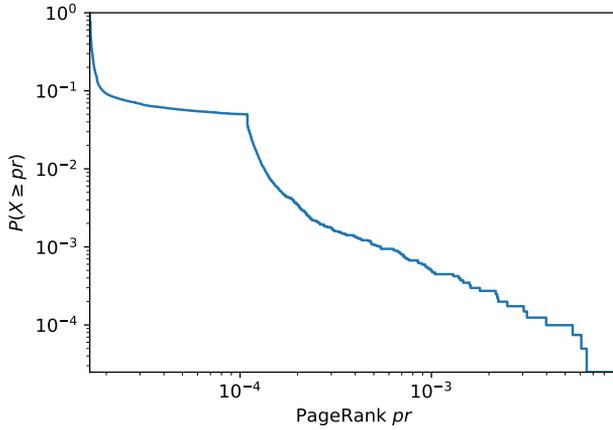
(c) Outdegree distribution



(d) Coreness distribution

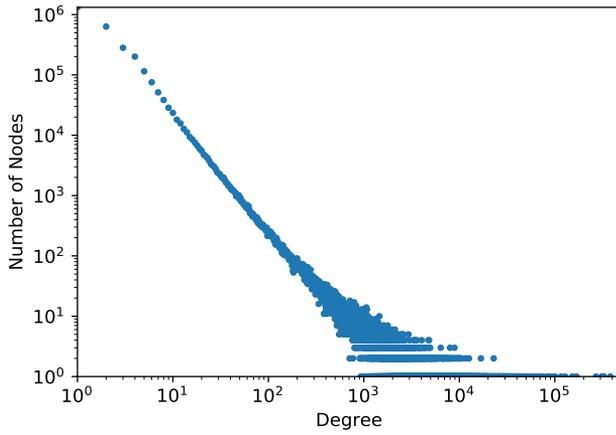


(e) Local clustering coefficient distribution

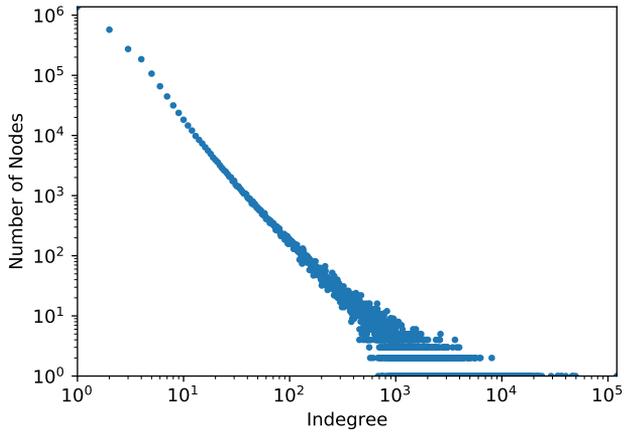


(f) PageRank distribution

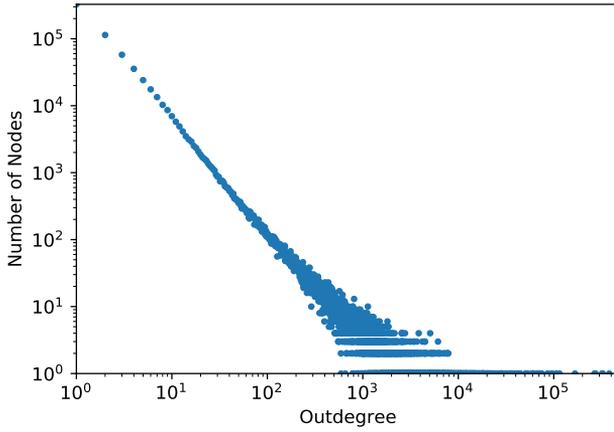
Figure B.4: The base feature distribution of the Wiki-talk-el network.



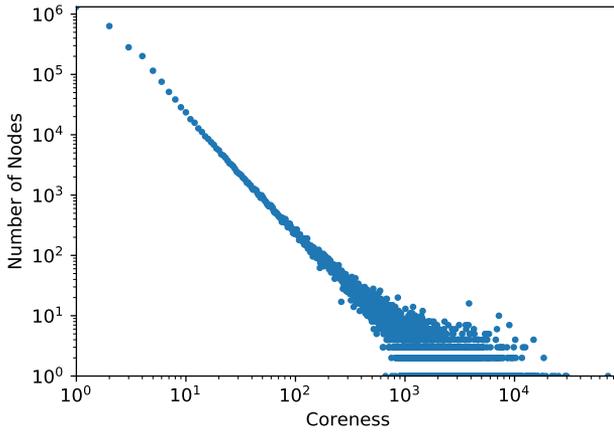
(a) Degree distribution



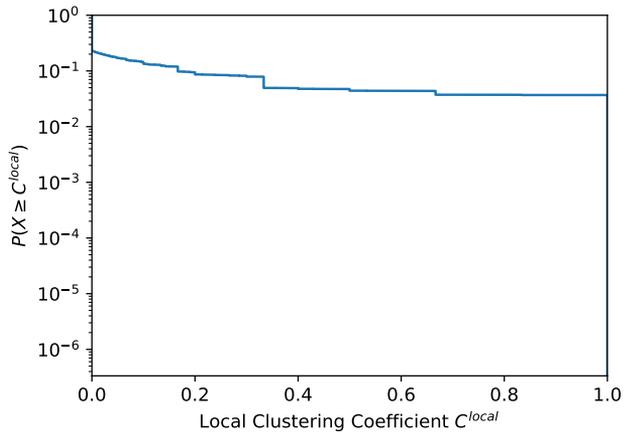
(b) Indegree distribution



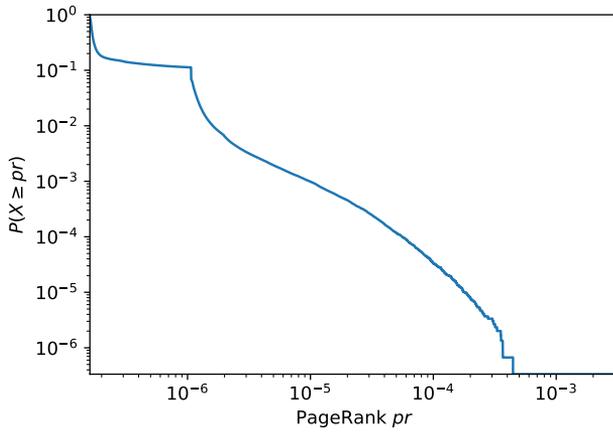
(c) Outdegree distribution



(d) Coreness distribution

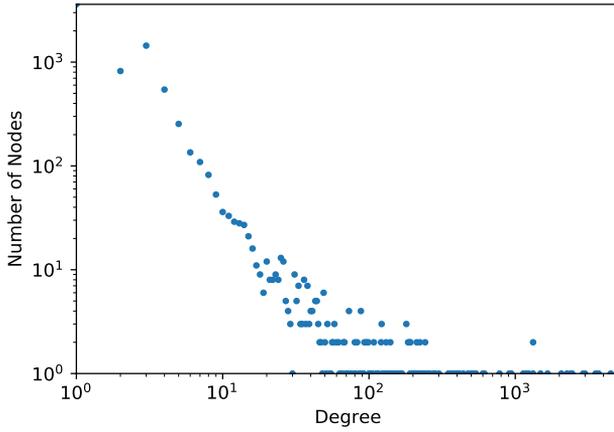


(e) Local clustering coefficient distribution

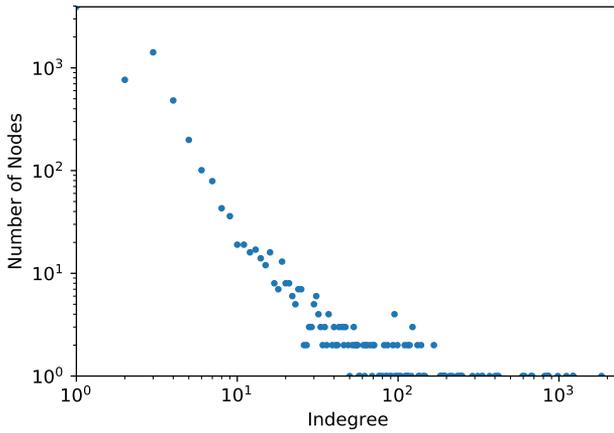


(f) PageRank distribution

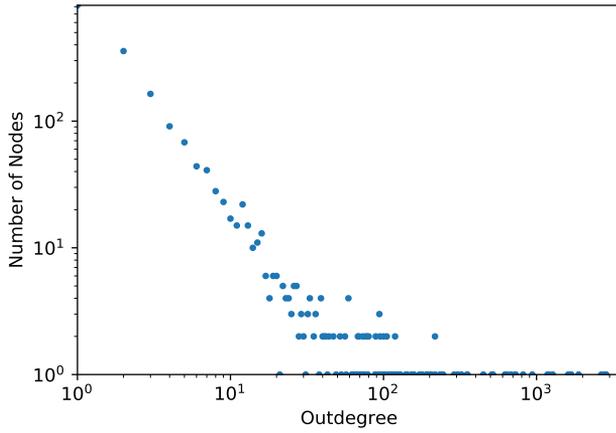
Figure B.5: The base feature distribution of the Wiki-talk-en network.



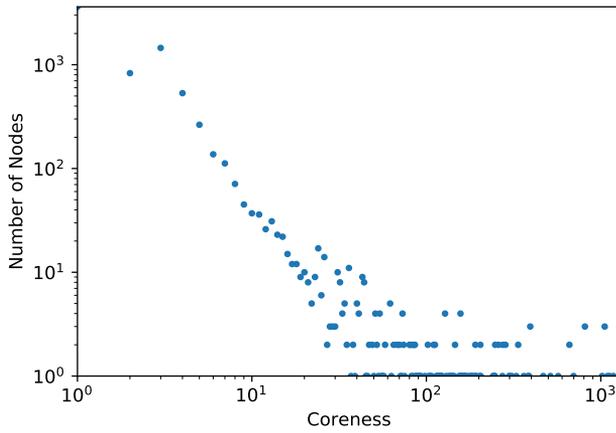
(a) Degree distribution



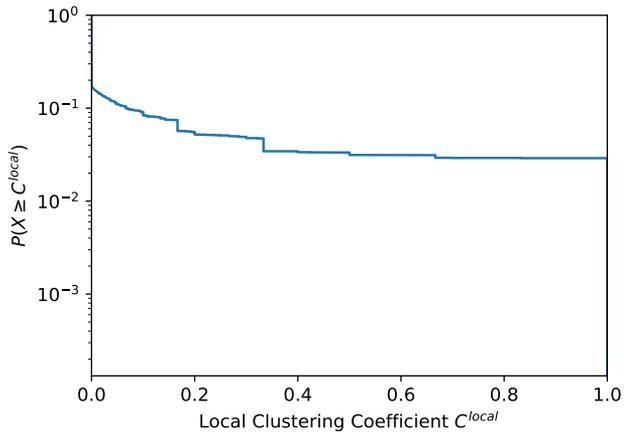
(b) Indegree distribution



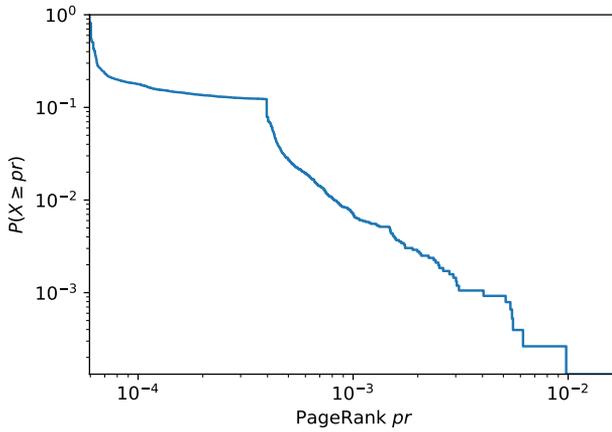
(c) Outdegree distribution



(d) Coreness distribution

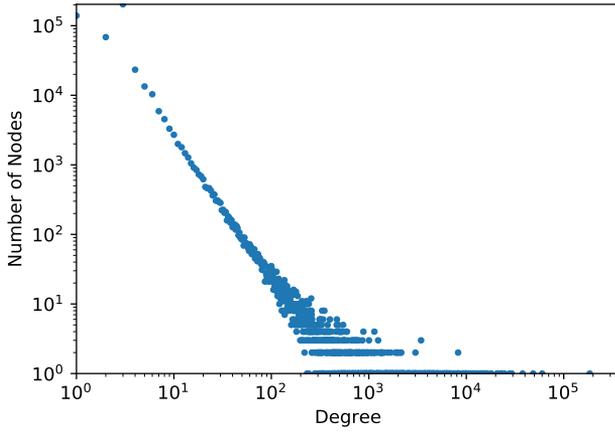


(e) Local clustering coefficient distribution

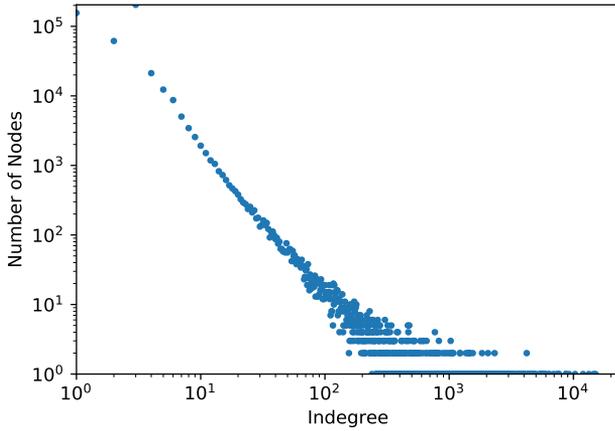


(f) PageRank distribution

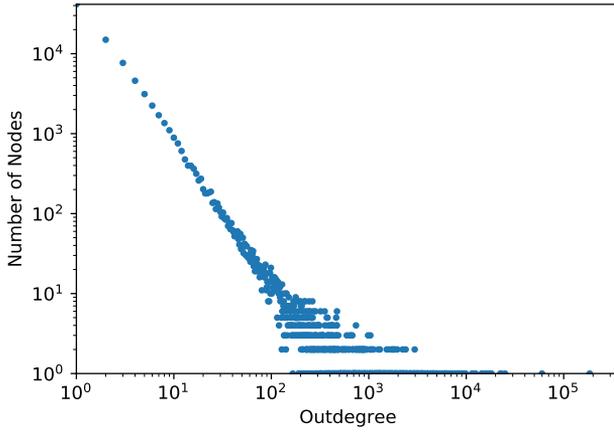
Figure B.6: The base feature distribution of the Wiki-talk-*eo* network.



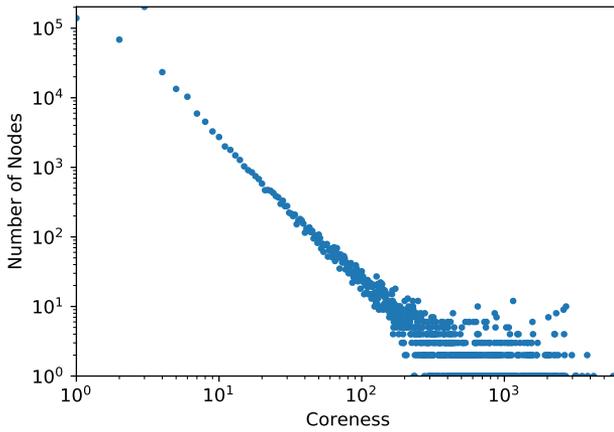
(a) Degree distribution



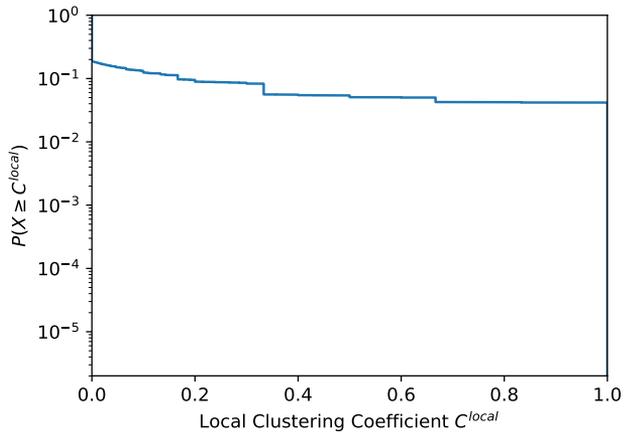
(b) Indegree distribution



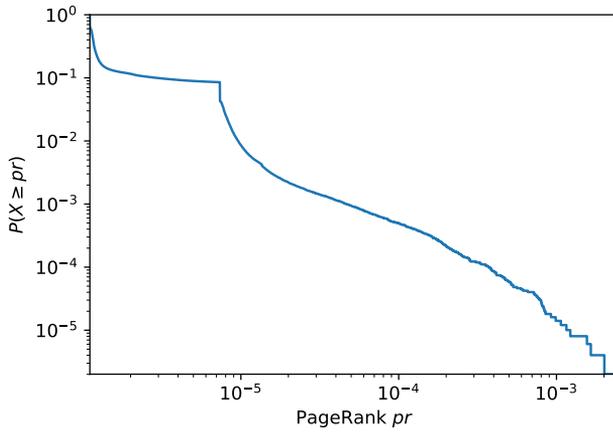
(c) Outdegree distribution



(d) Coreness distribution

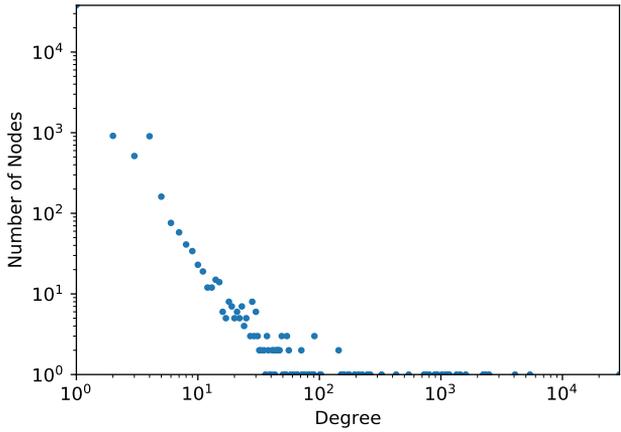


(e) Local clustering coefficient distribution

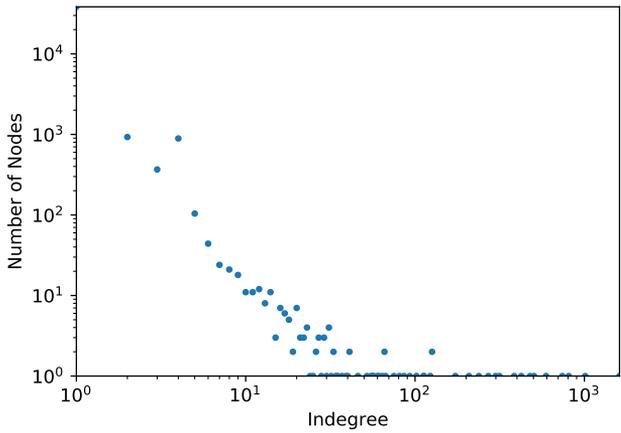


(f) PageRank distribution

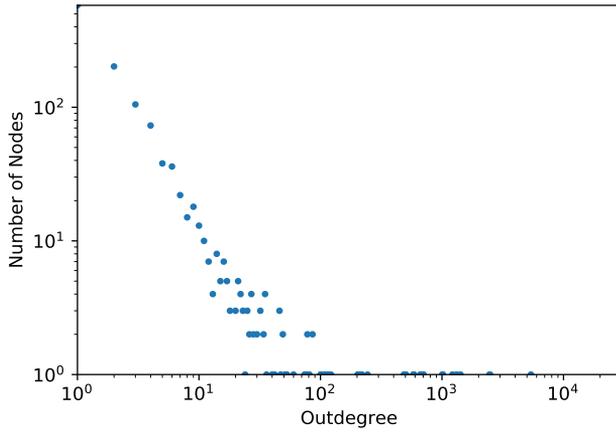
Figure B.7: The base feature distribution of the Wiki-talk-es network.



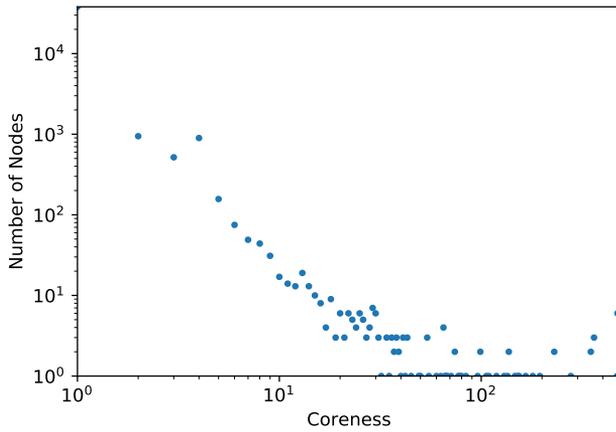
(a) Degree distribution



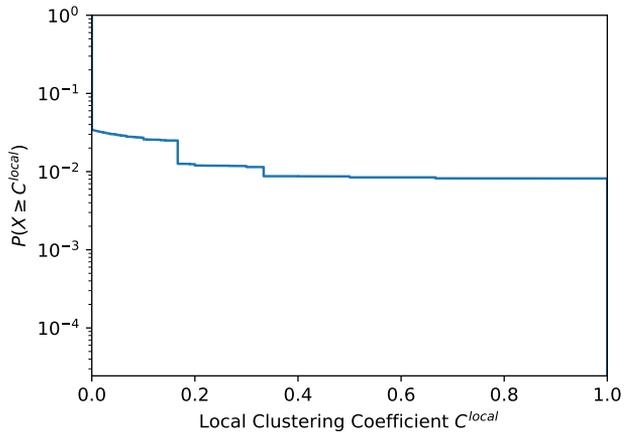
(b) Indegree distribution



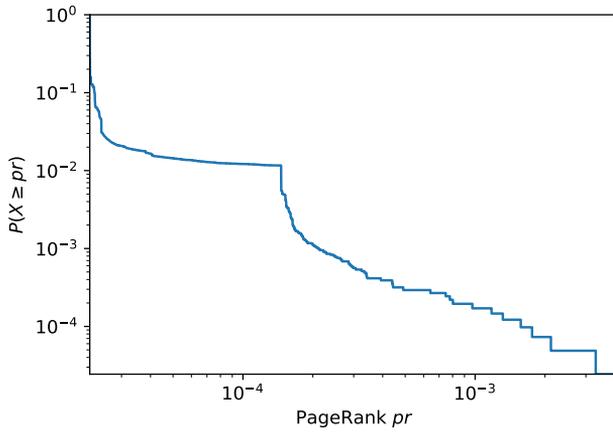
(c) Outdegree distribution



(d) Coreness distribution

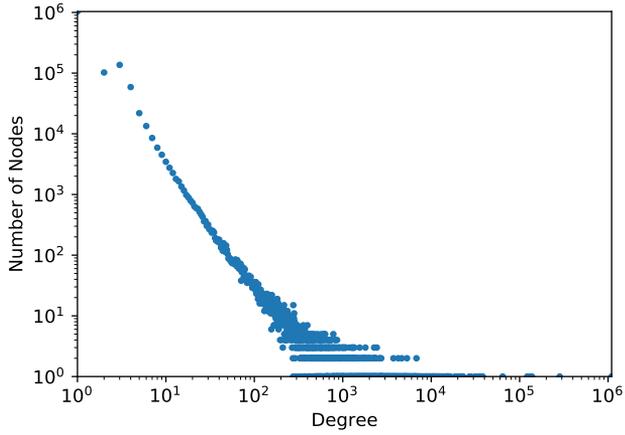


(e) Local clustering coefficient distribution

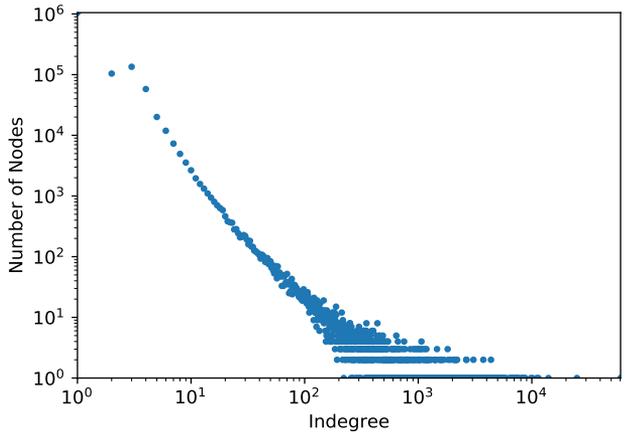


(f) PageRank distribution

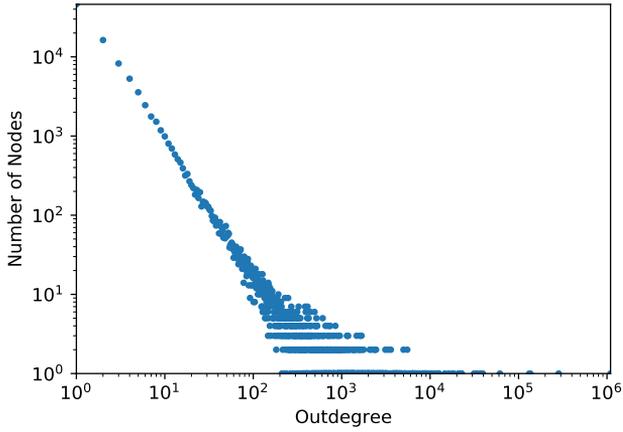
Figure B.8: The base feature distribution of the Wiki-talk-eu network.



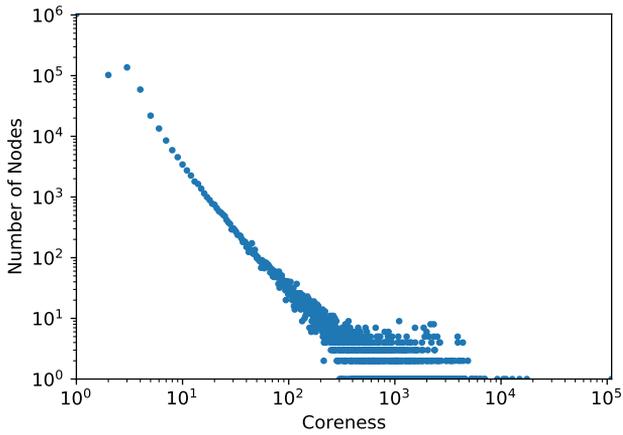
(a) Degree distribution



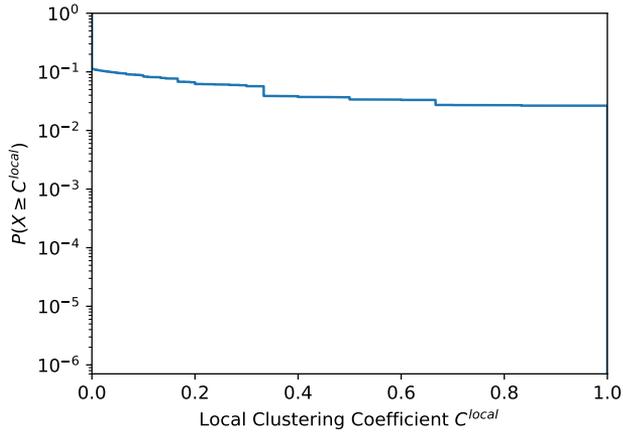
(b) Indegree distribution



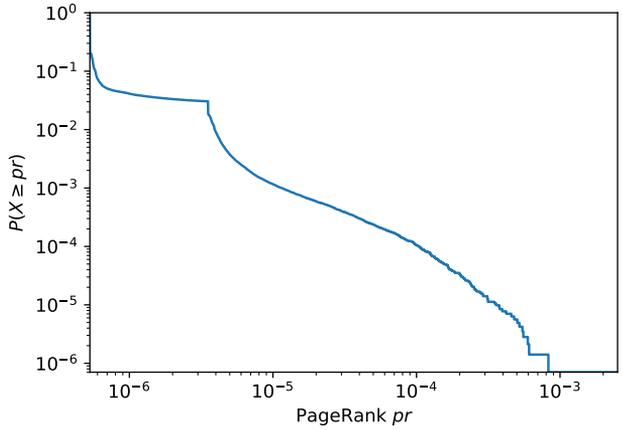
(c) Outdegree distribution



(d) Coreness distribution

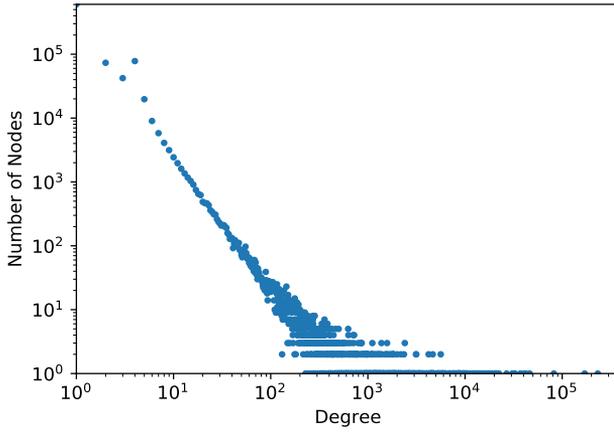


(e) Local clustering coefficient distribution

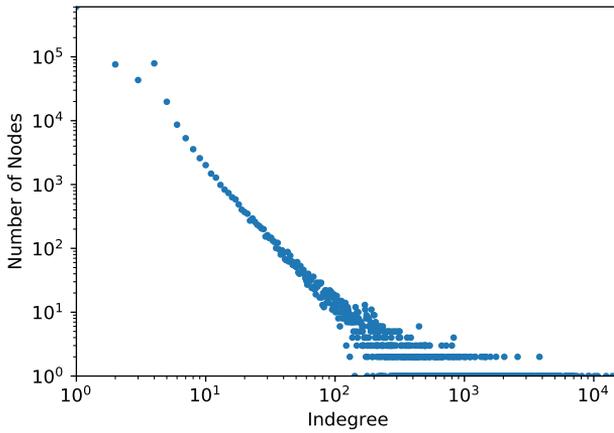


(f) PageRank distribution

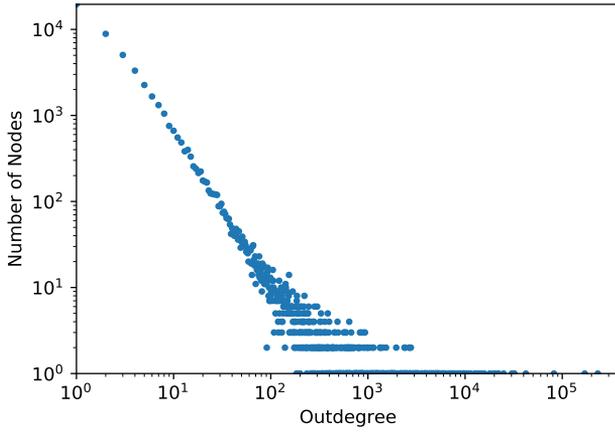
Figure B.9: The base feature distribution of the Wiki-talk-fr network.



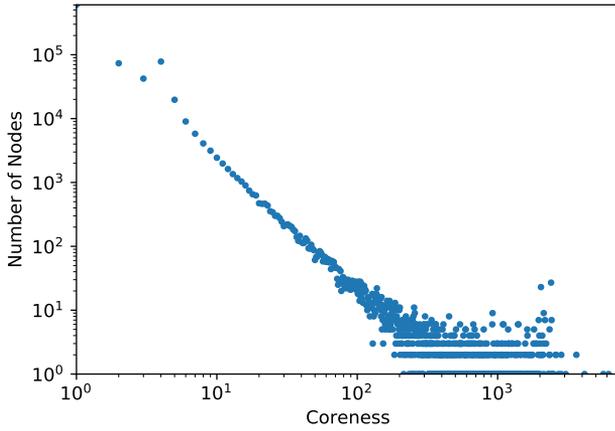
(a) Degree distribution



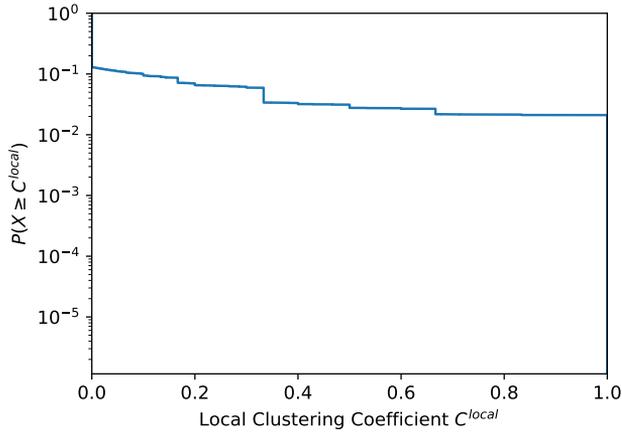
(b) Indegree distribution



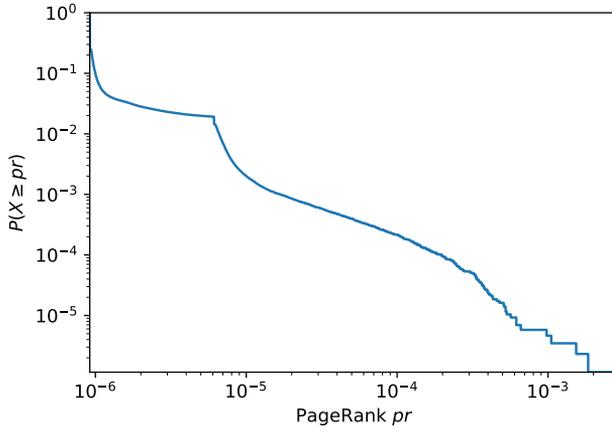
(c) Outdegree distribution



(d) Coreness distribution

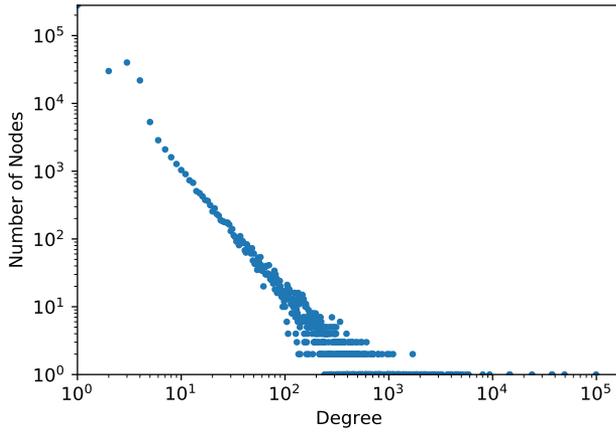


(e) Local clustering coefficient distribution

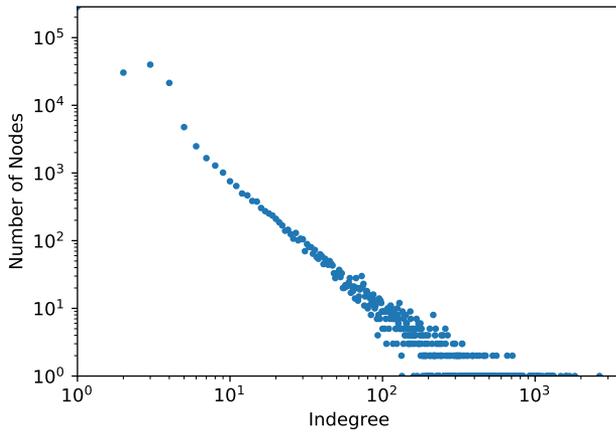


(f) PageRank distribution

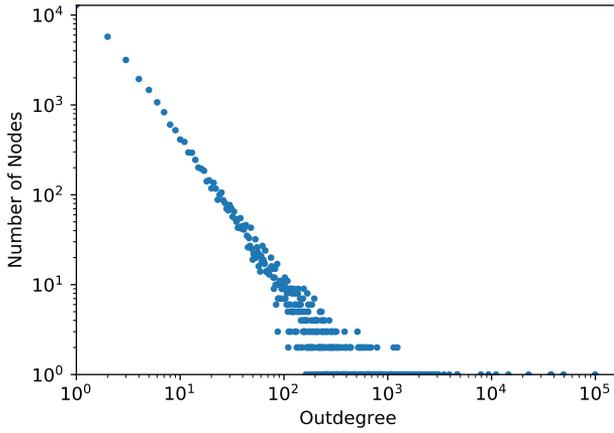
Figure B.10: The base feature distribution of the Wiki-talk-it network.



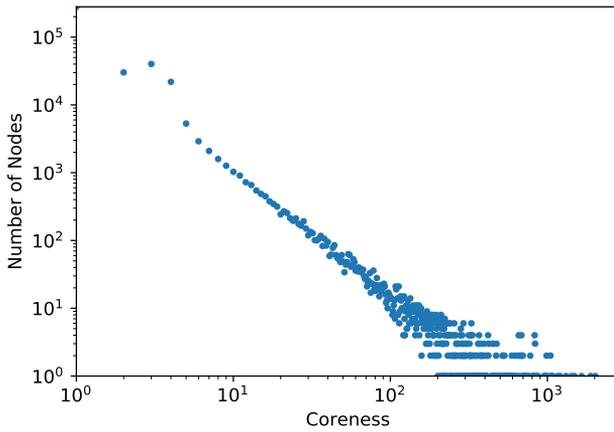
(a) Degree distribution



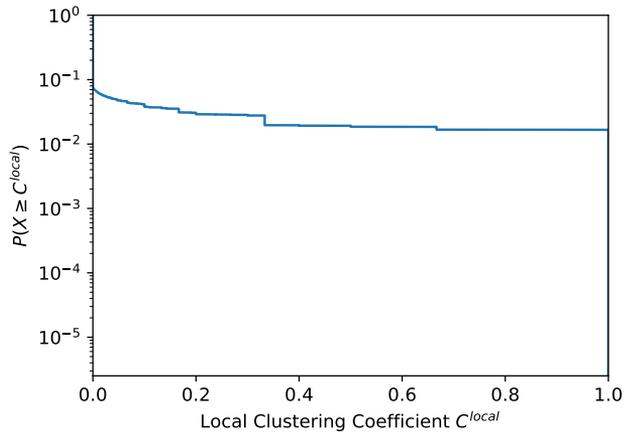
(b) Indegree distribution



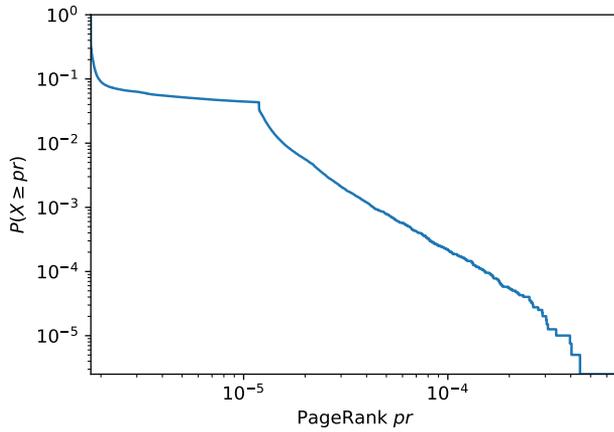
(c) Outdegree distribution



(d) Coreness distribution

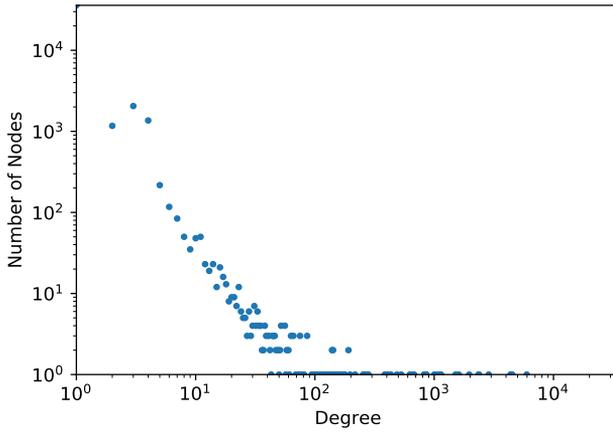


(e) Local clustering coefficient distribution

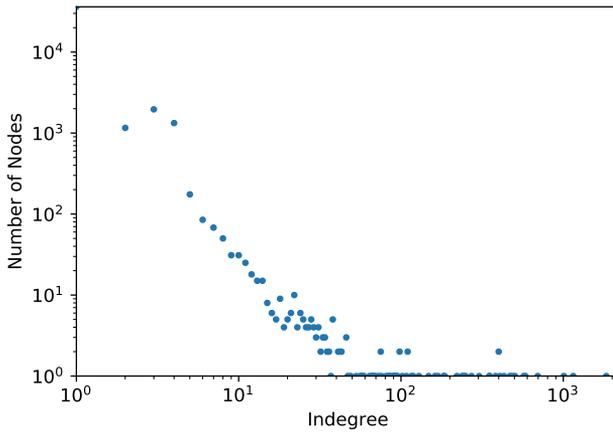


(f) PageRank distribution

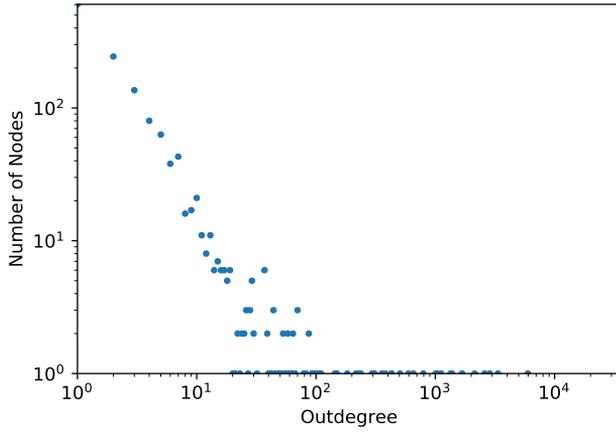
Figure B.11: The base feature distribution of the Wiki-talk-ja network.



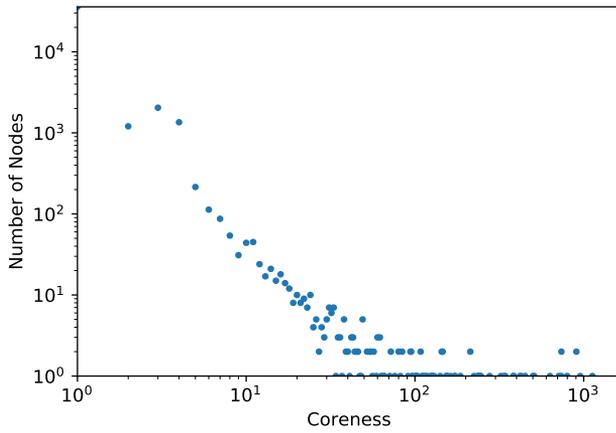
(a) Degree distribution



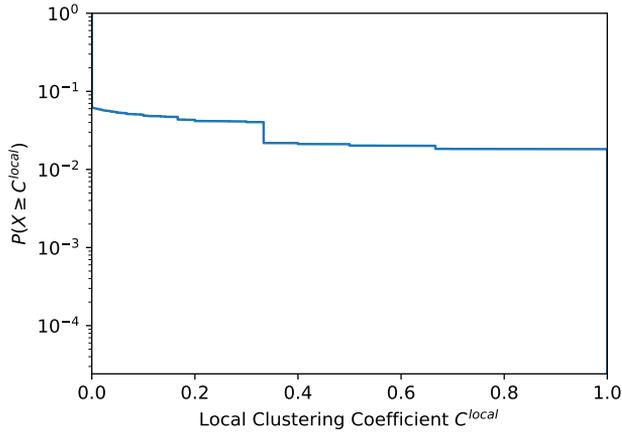
(b) Indegree distribution



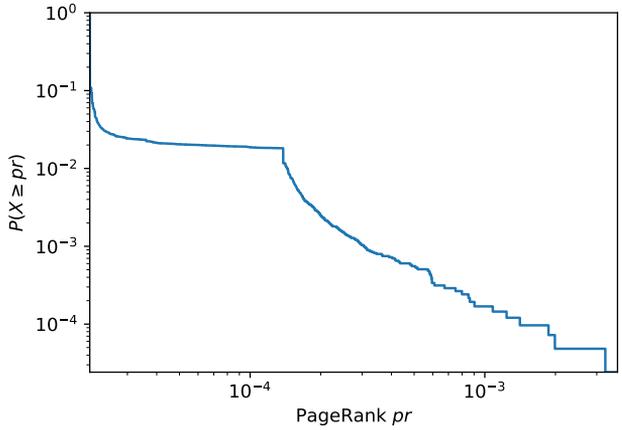
(c) Outdegree distribution



(d) Coreness distribution

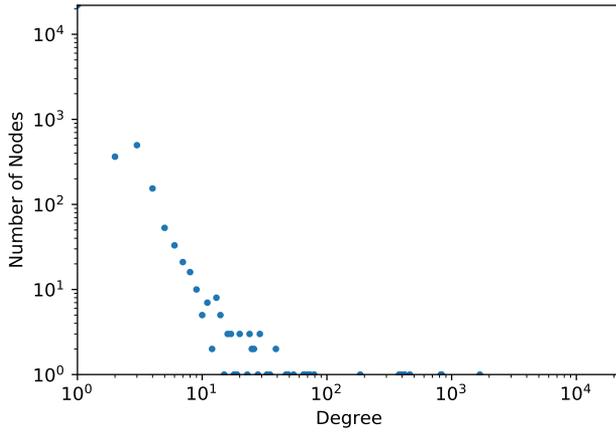


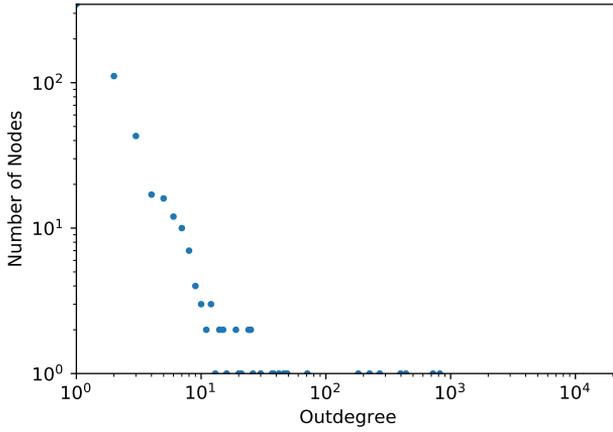
(e) Local clustering coefficient distribution

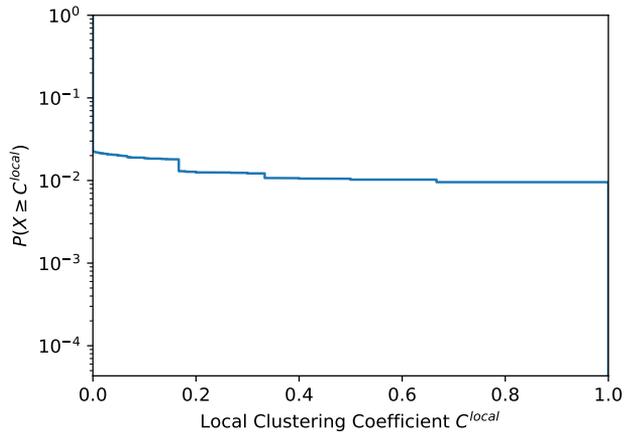


(f) PageRank distribution

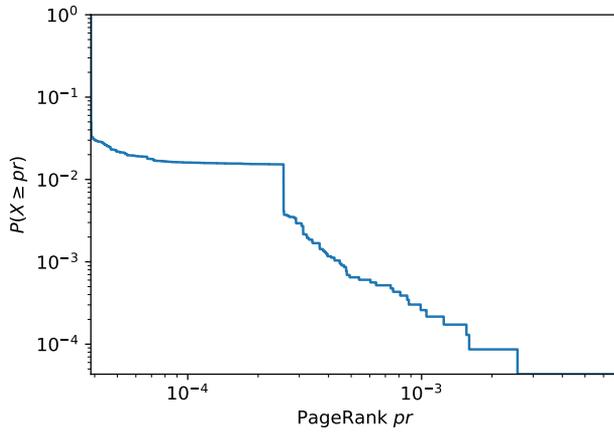
Figure B.12: The base feature distribution of the Wiki-talk-iv network.





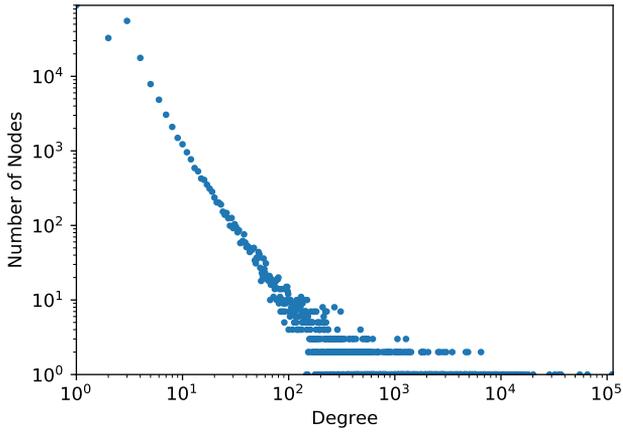


(e) Local clustering coefficient distribution

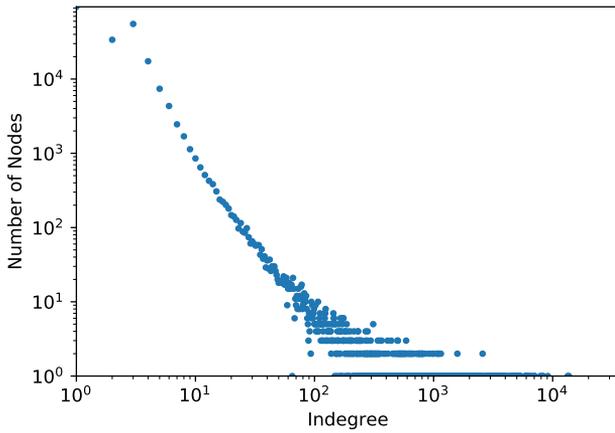


(f) PageRank distribution

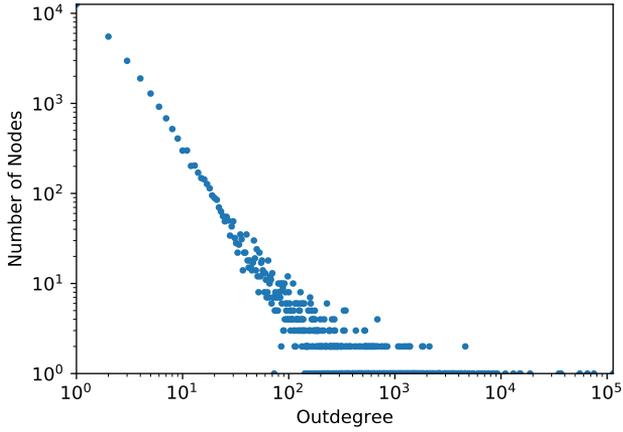
Figure B.13: The base feature distribution of the Wiki-talk-nds network.



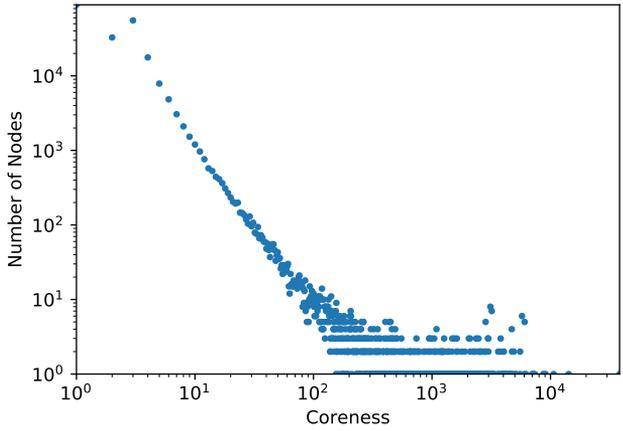
(a) Degree distribution



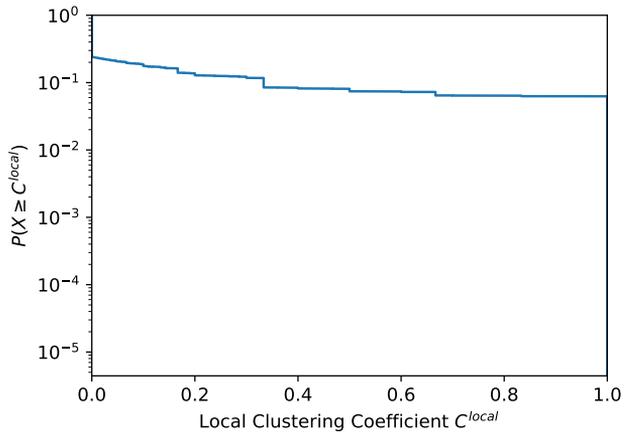
(b) Indegree distribution



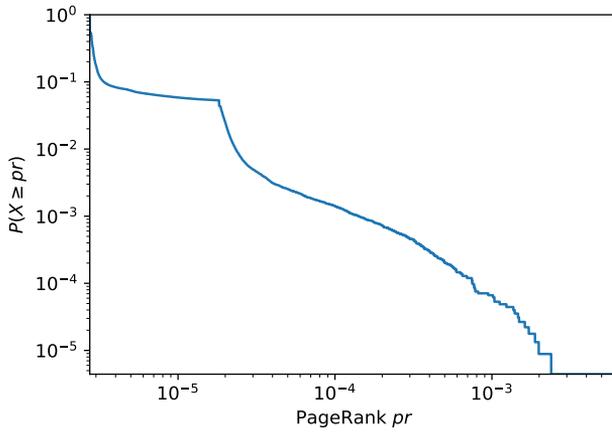
(c) Outdegree distribution



(d) Coreness distribution

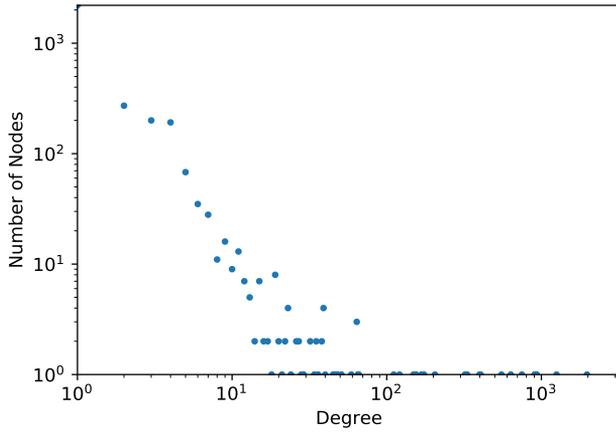


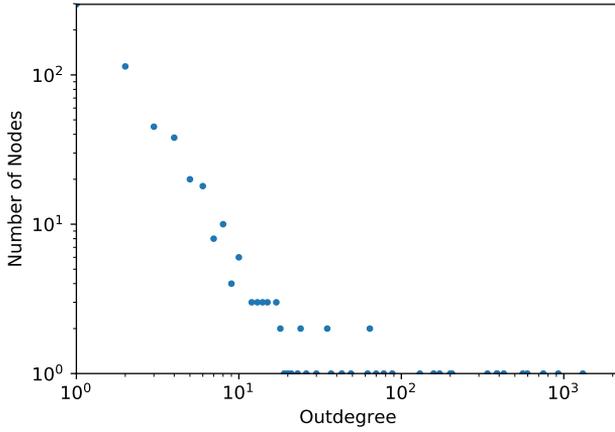
(e) Local clustering coefficient distribution



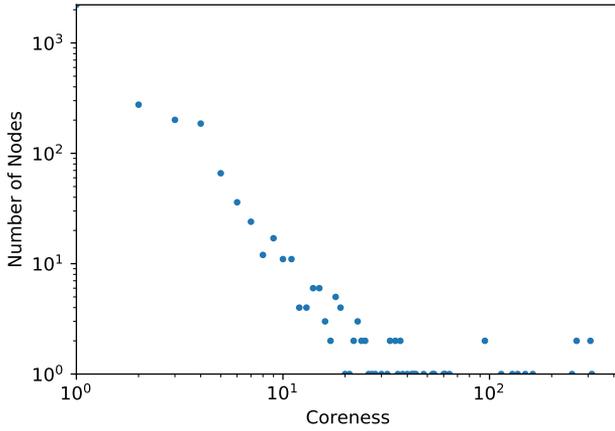
(f) PageRank distribution

Figure B.14: The base feature distribution of the Wiki-talk-nl network.

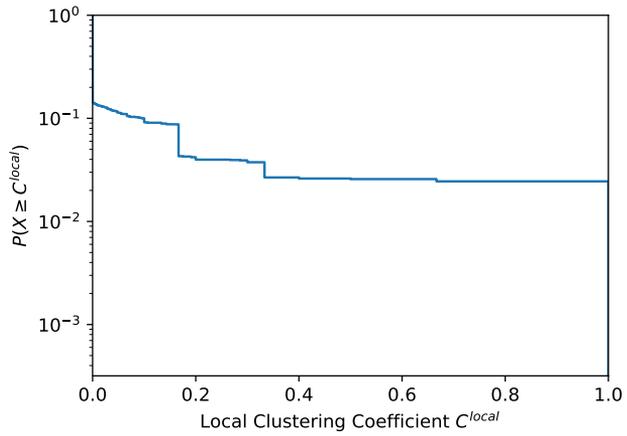




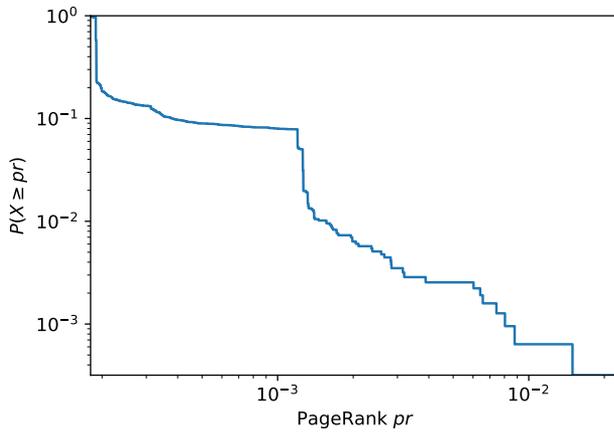
(c) Outdegree distribution



(d) Coreness distribution

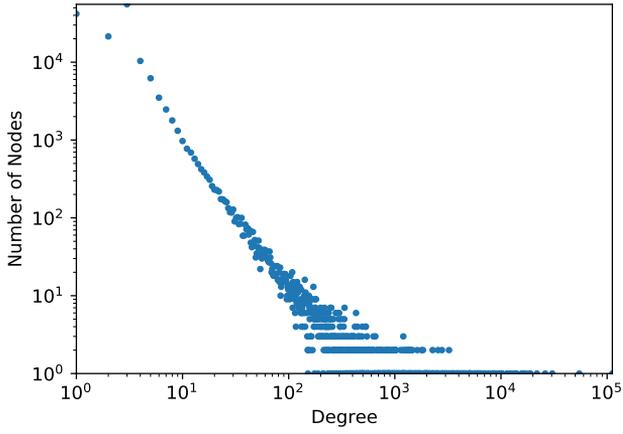


(e) Local clustering coefficient distribution

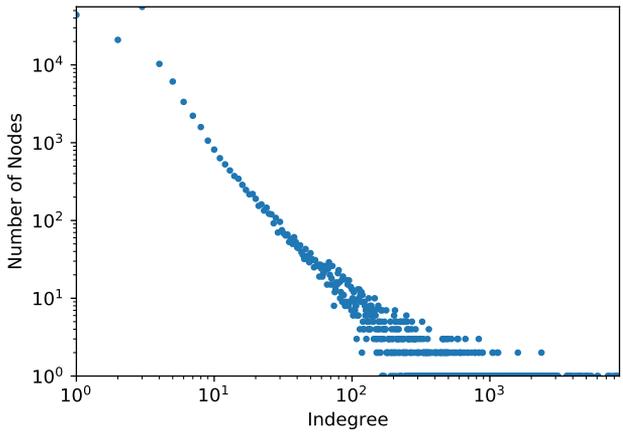


(f) PageRank distribution

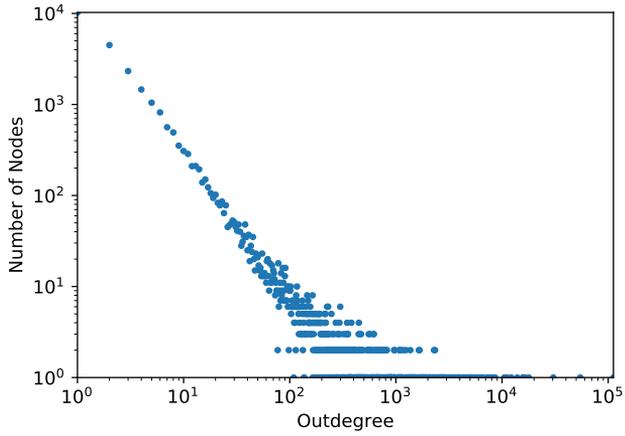
Figure B.15: The base feature distribution of the Wiki-talk-oc network.



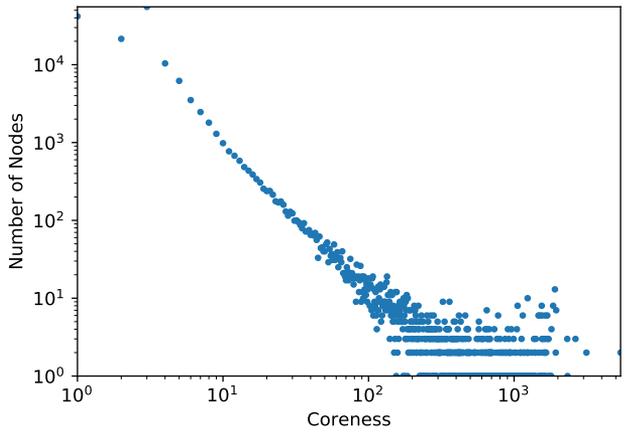
(a) Degree distribution



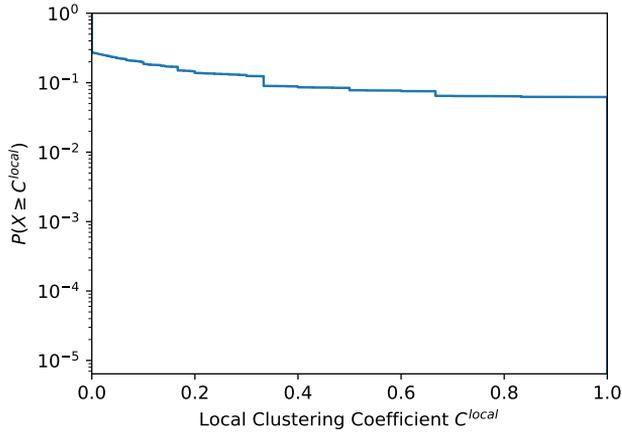
(b) Indegree distribution



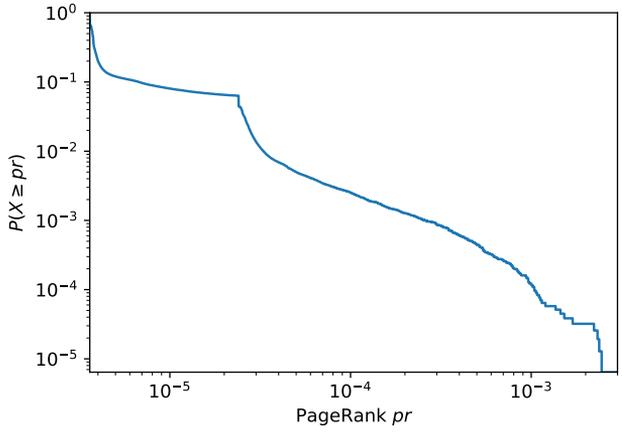
(c) Outdegree distribution



(d) Coreness distribution

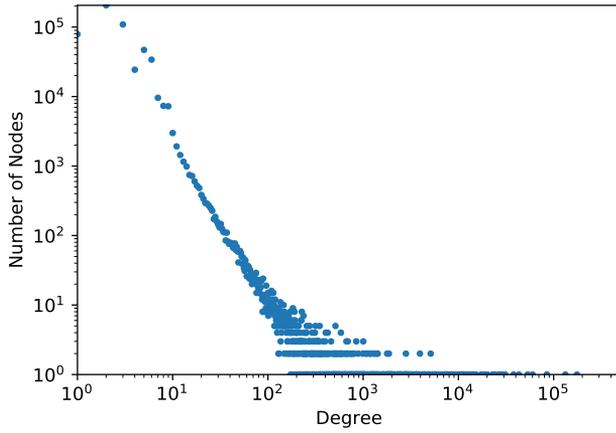


(e) Local clustering coefficient distribution

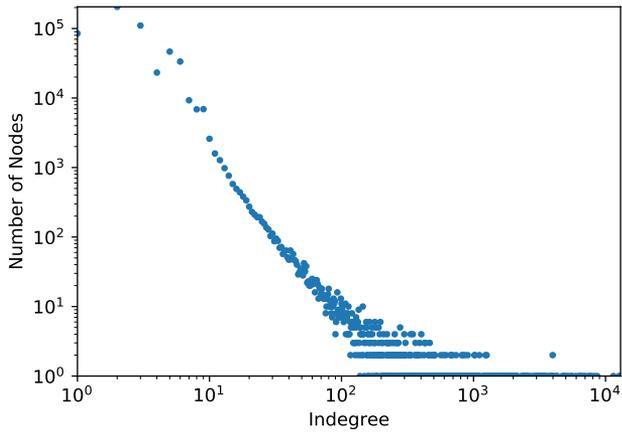


(f) PageRank distribution

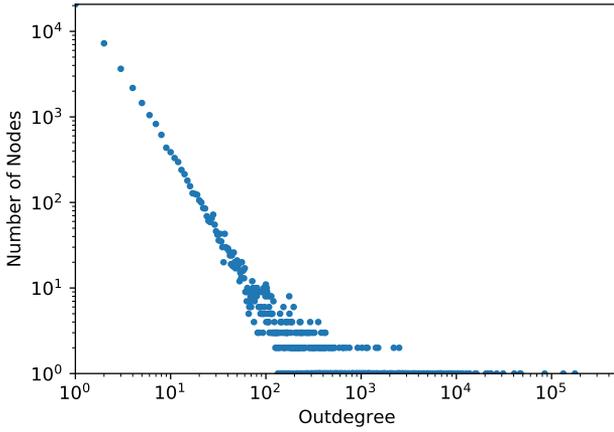
Figure B.16: The base feature distribution of the Wiki-talk-pl network.



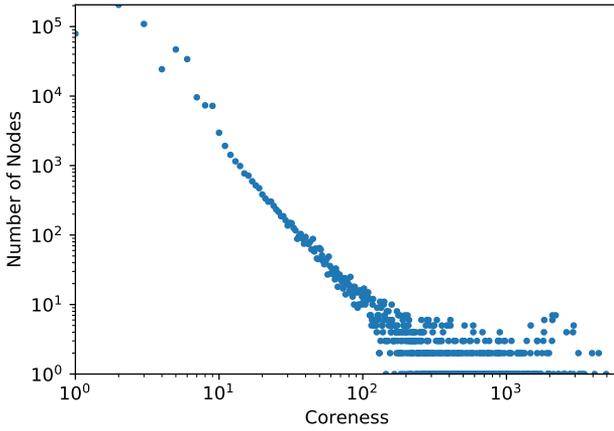
(a) Degree distribution



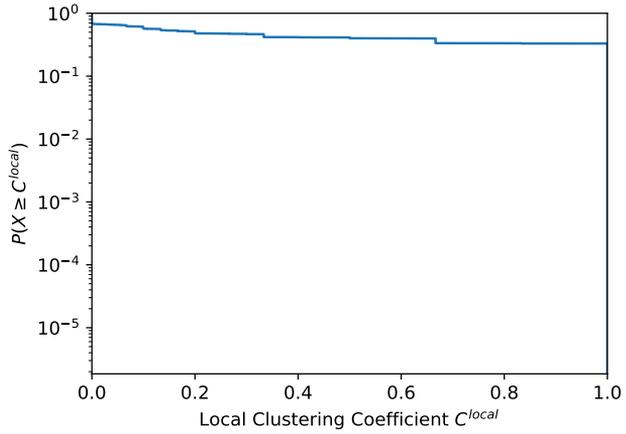
(b) Indegree distribution



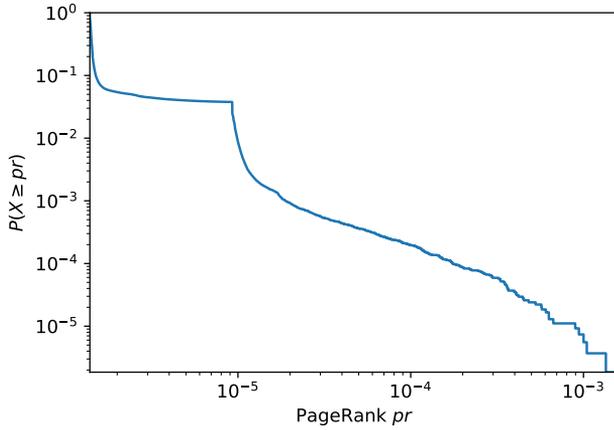
(c) Outdegree distribution



(d) Coreness distribution

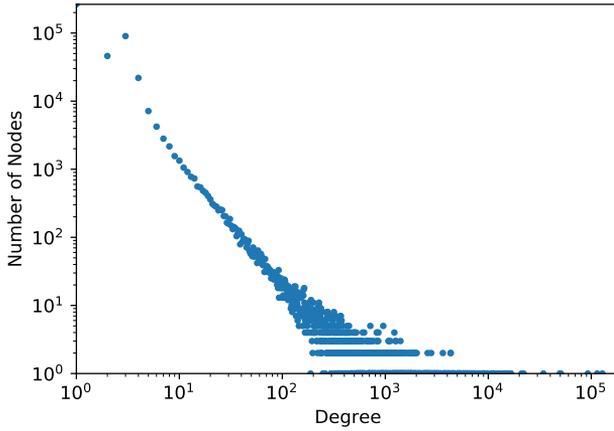


(e) Local clustering coefficient distribution

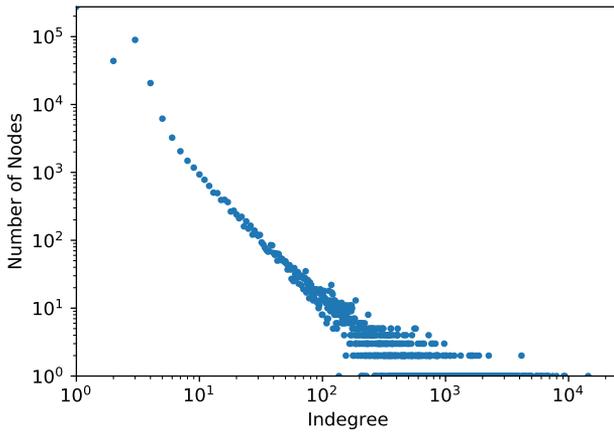


(f) PageRank distribution

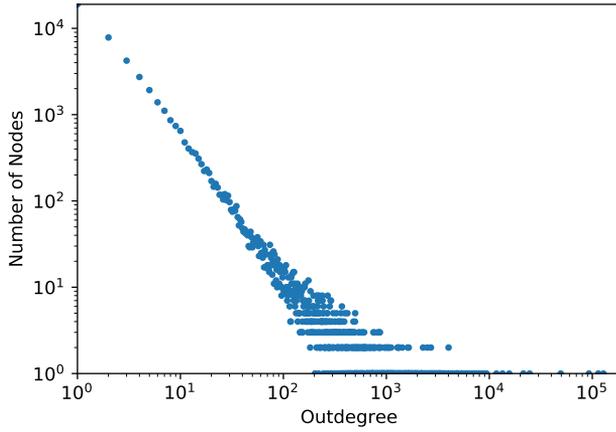
Figure B.17: The base feature distribution of the Wiki-talk-pt network.



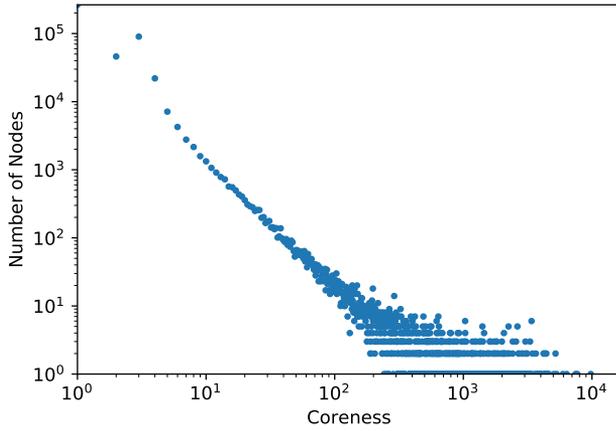
(a) Degree distribution



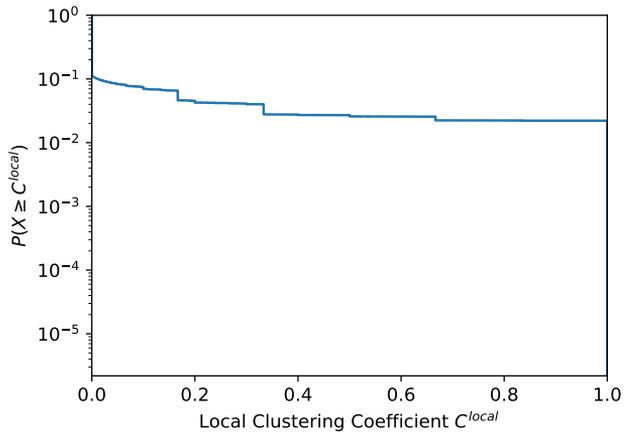
(b) Indegree distribution



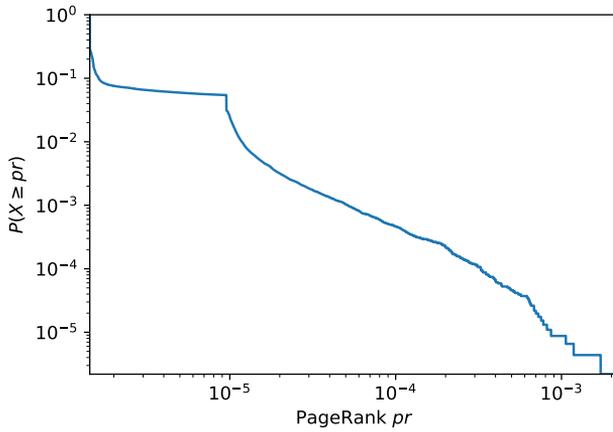
(c) Outdegree distribution



(d) Coreness distribution

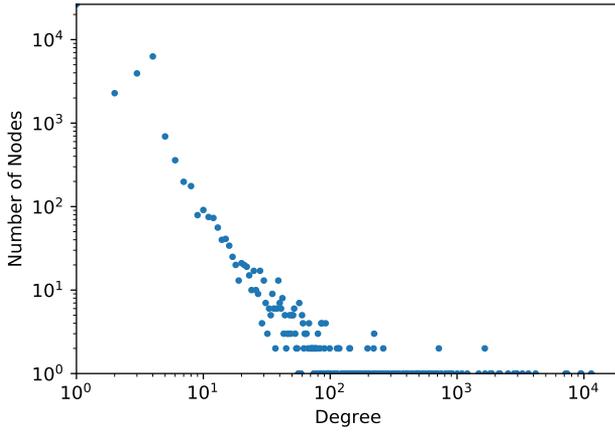


(e) Local clustering coefficient distribution

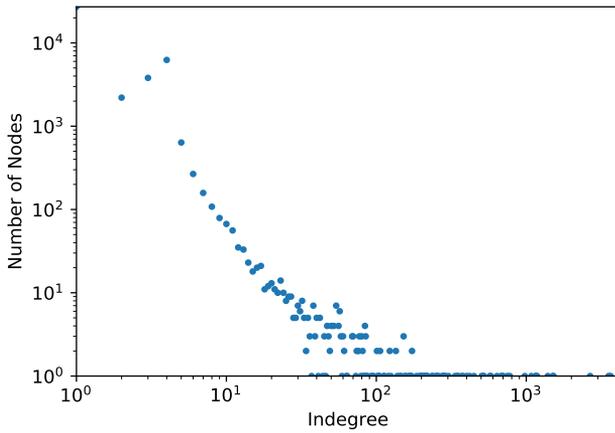


(f) PageRank distribution

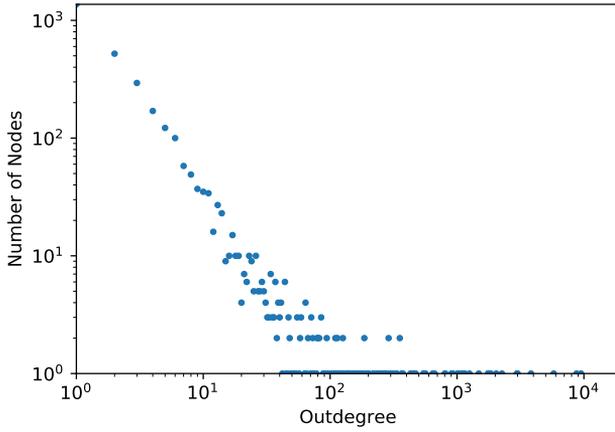
Figure B.18: The base feature distribution of the Wiki-talk-ru network.



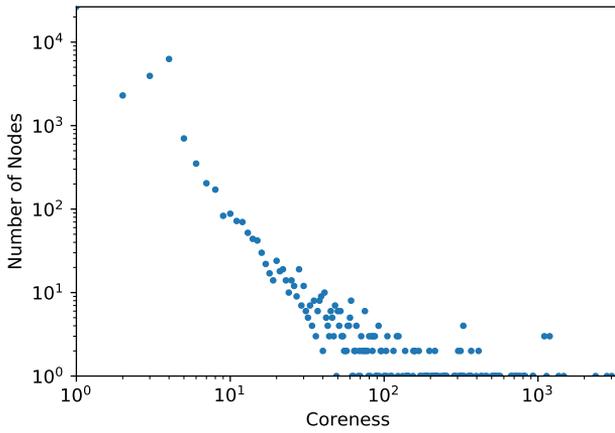
(a) Degree distribution



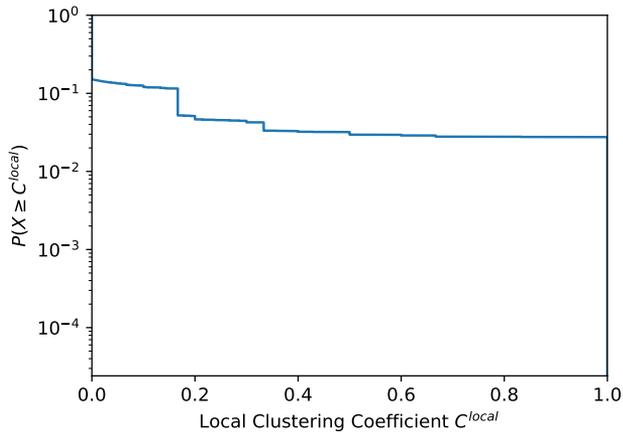
(b) Indegree distribution



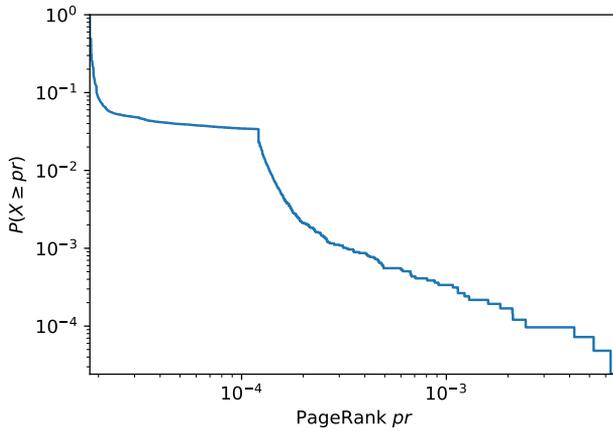
(c) Outdegree distribution



(d) Coreness distribution

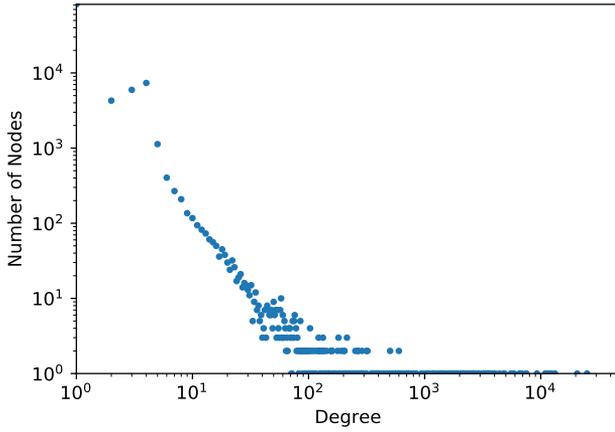


(e) Local clustering coefficient distribution

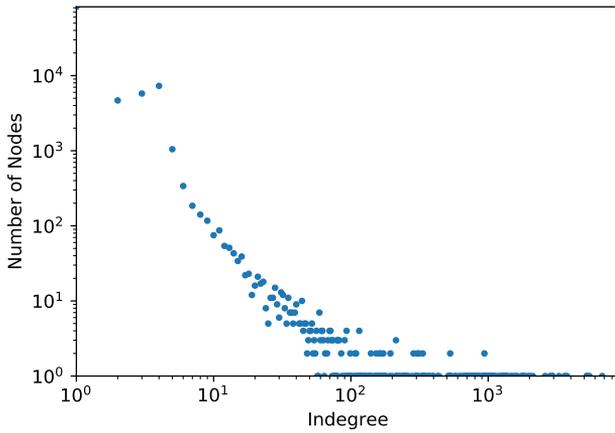


(f) PageRank distribution

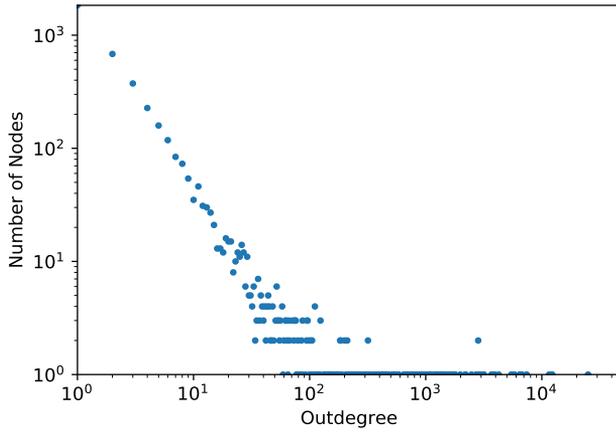
Figure B.19: The base feature distribution of the Wiki-talk-sk network.



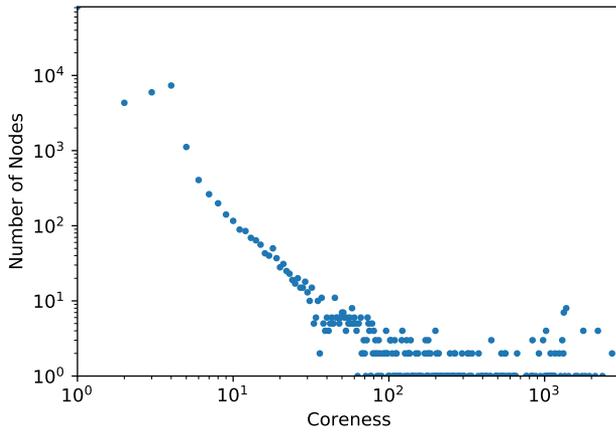
(a) Degree distribution



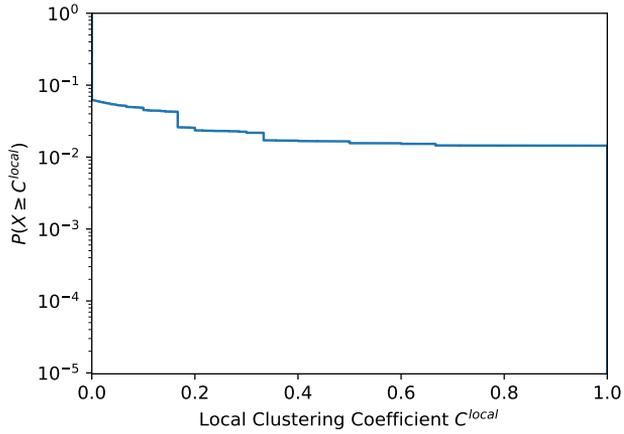
(b) Indegree distribution



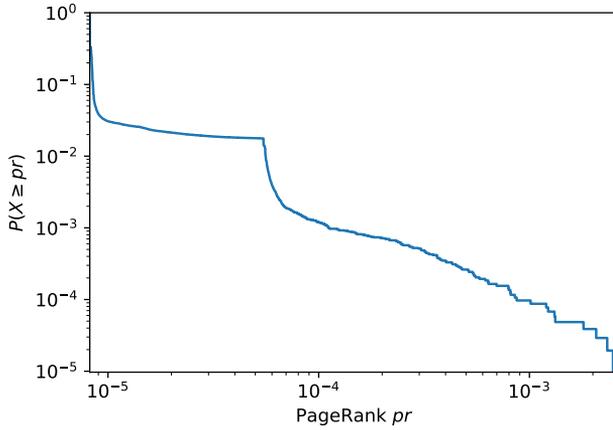
(c) Outdegree distribution



(d) Coreness distribution

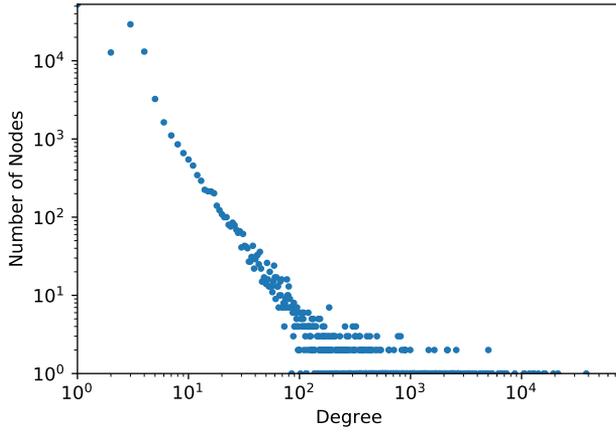


(e) Local clustering coefficient distribution

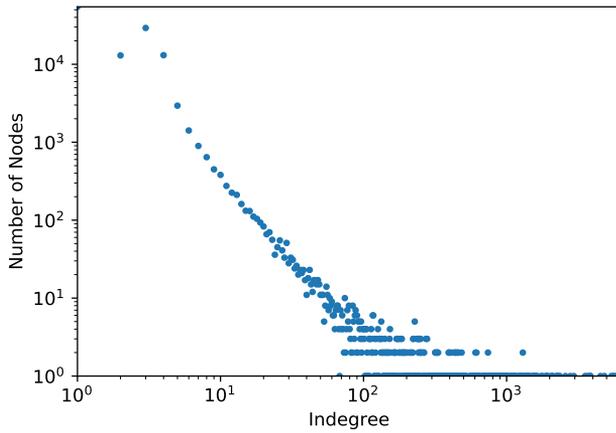


(f) PageRank distribution

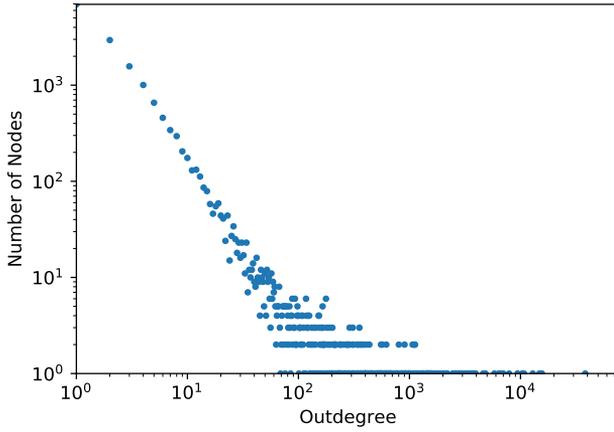
Figure B.20: The base feature distribution of the Wiki-talk-sr network.



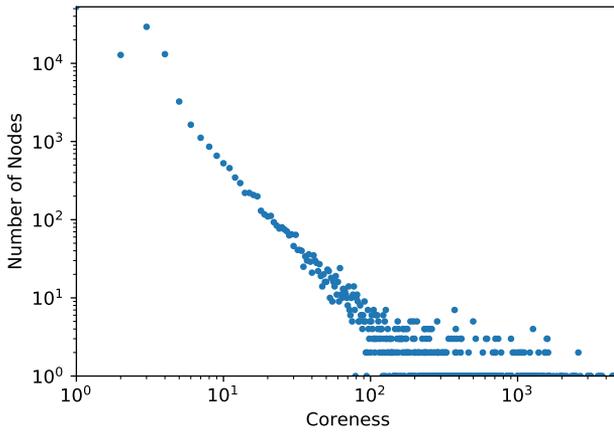
(a) Degree distribution



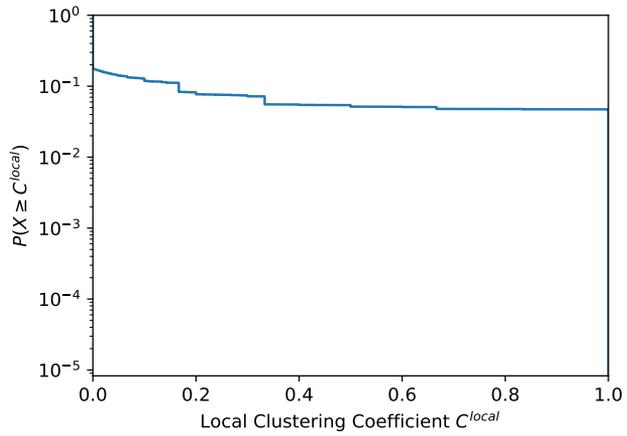
(b) Indegree distribution



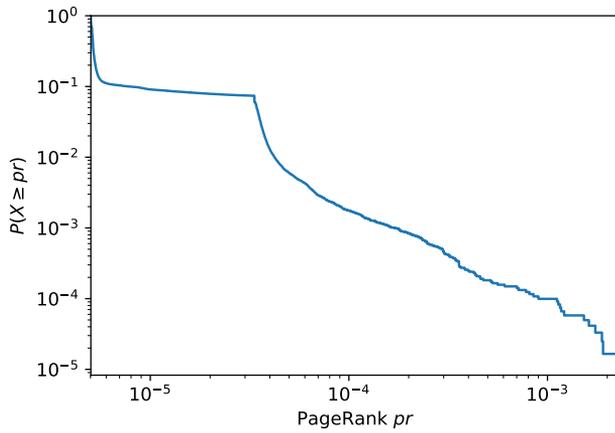
(c) Outdegree distribution



(d) Coreness distribution

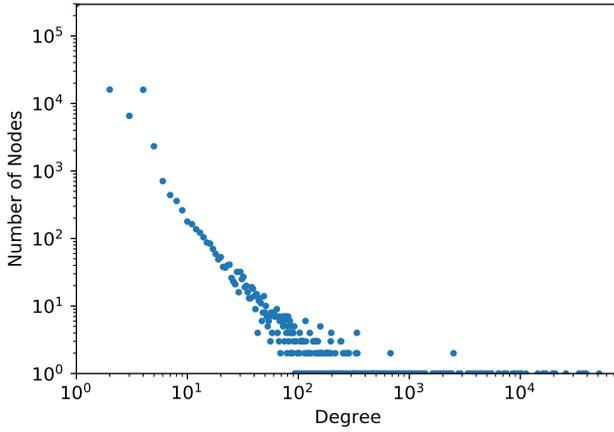


(e) Local clustering coefficient distribution

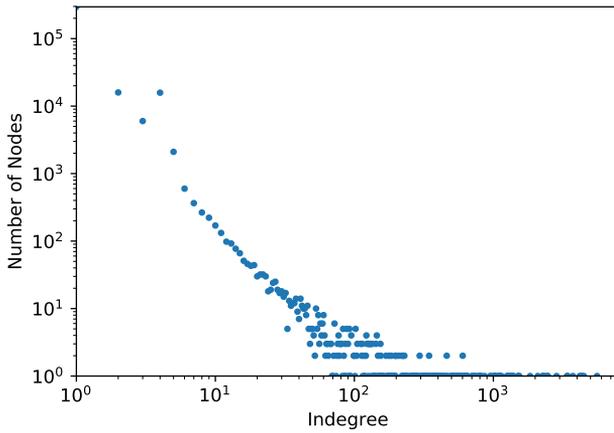


(f) PageRank distribution

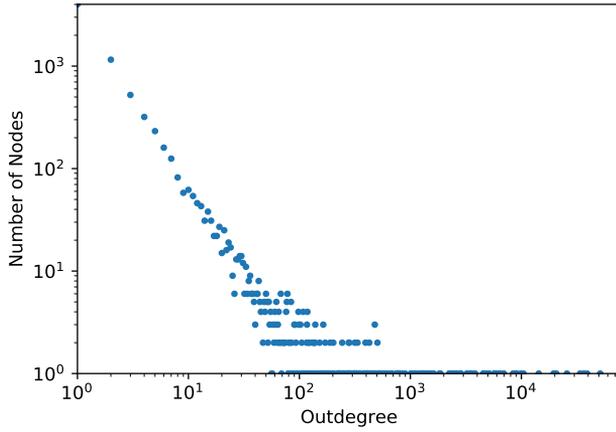
Figure B.21: The base feature distribution of the Wiki-talk-sv network.



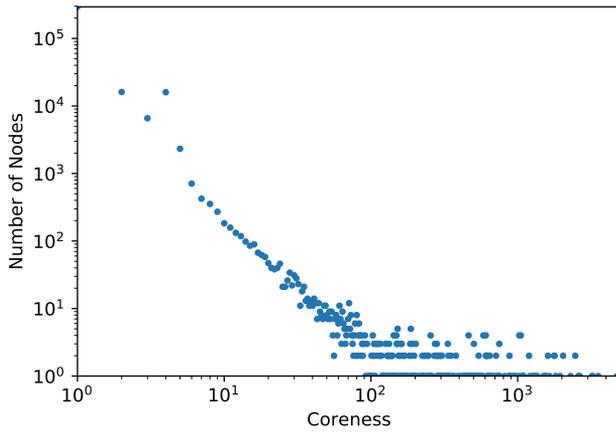
(a) Degree distribution



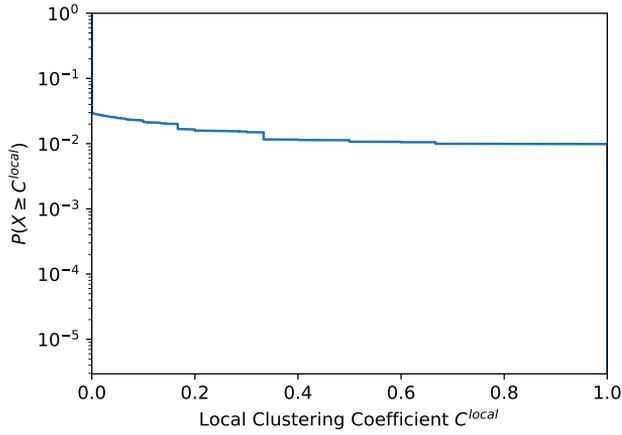
(b) Indegree distribution



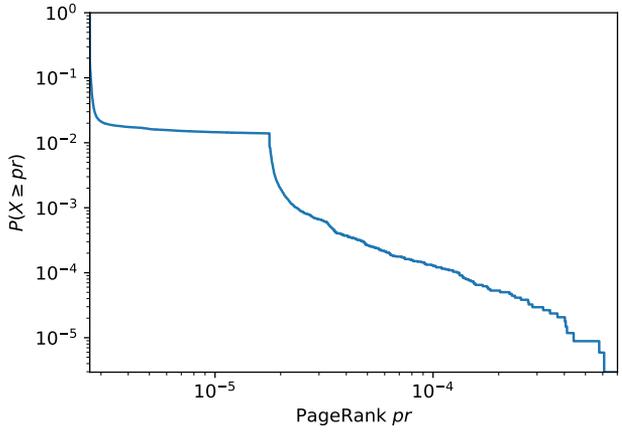
(c) Outdegree distribution



(d) Coreness distribution

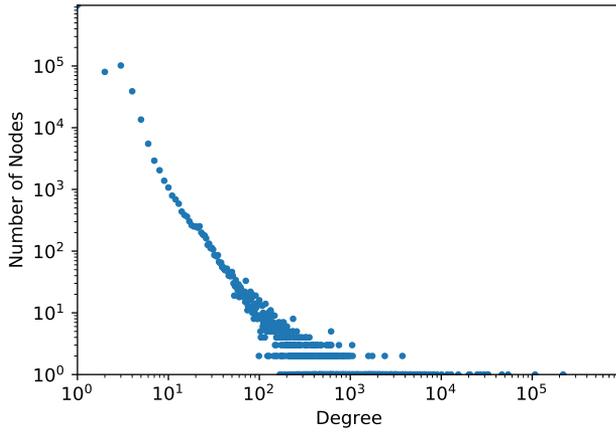


(e) Local clustering coefficient distribution

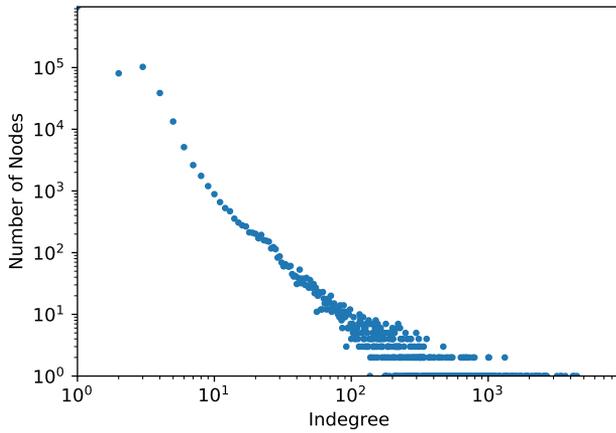


(f) PageRank distribution

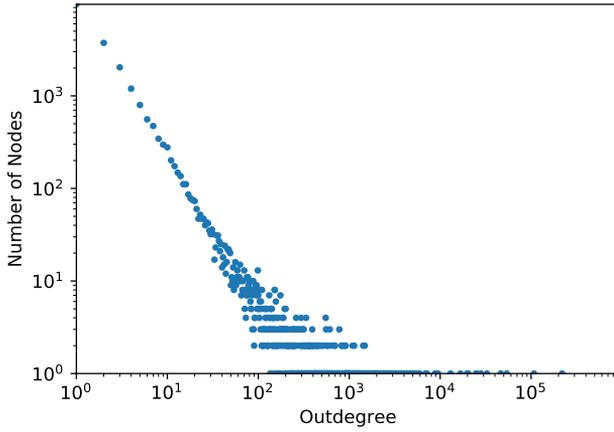
Figure B.22: The base feature distribution of the Wiki-talk-vi network.



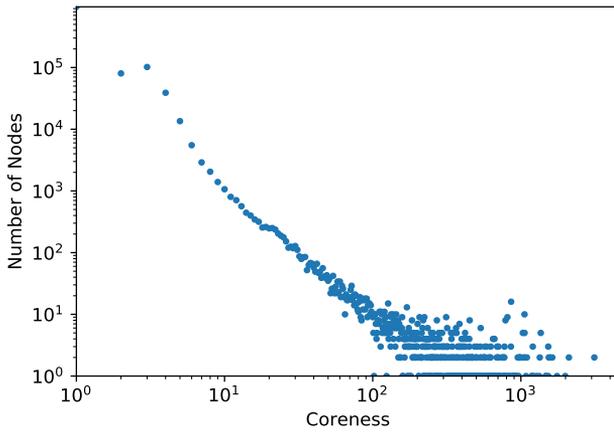
(a) Degree distribution



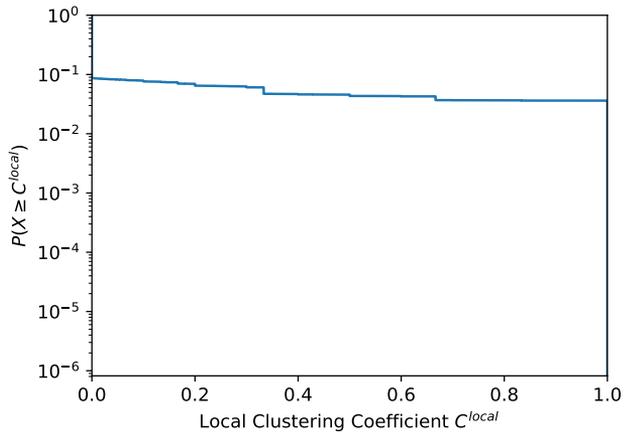
(b) Indegree distribution



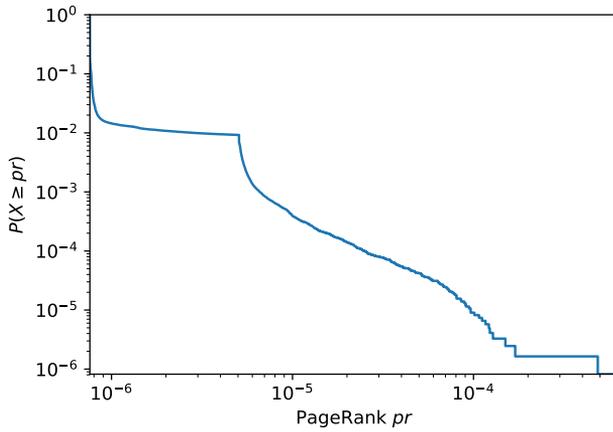
(c) Outdegree distribution



(d) Coreness distribution



(e) Local clustering coefficient distribution



(f) PageRank distribution

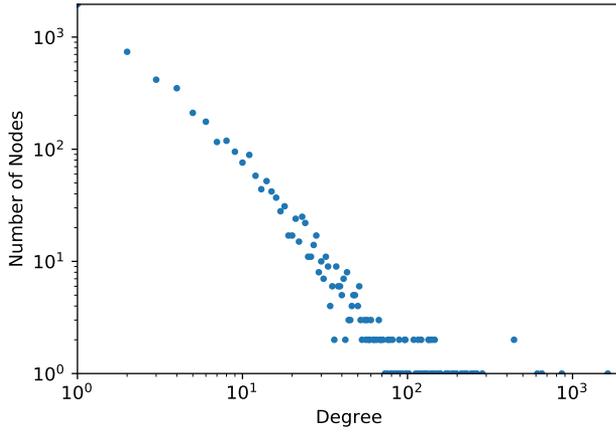
Figure B.23: The base feature distribution of the Wiki-talk-zh network.

B.2 BASE FEATURE DISTRIBUTIONS OF ARIS, SLASHDOT-ZOO AND EPINION-TRUST

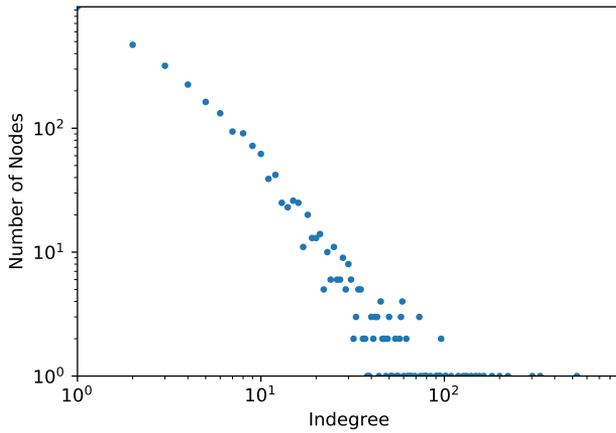
In Figure [B.24](#) we show the base feature distributions of nodes in the ARIS network.

In Figure [B.25](#) we show the base feature distributions of nodes in the Slashdot-Zoo network.

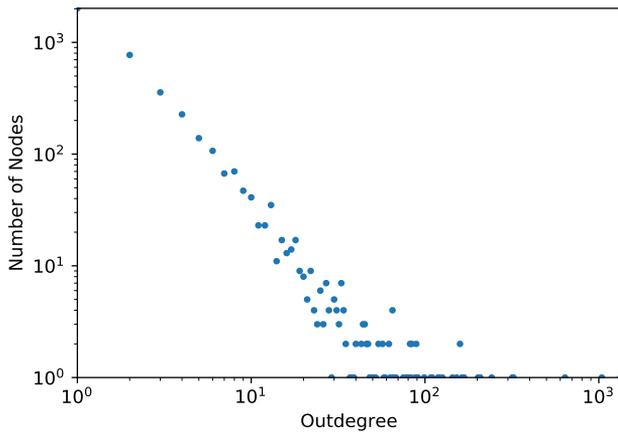
In Figure [B.26](#) we show the base feature distributions of nodes in the Epinion-Trust network.



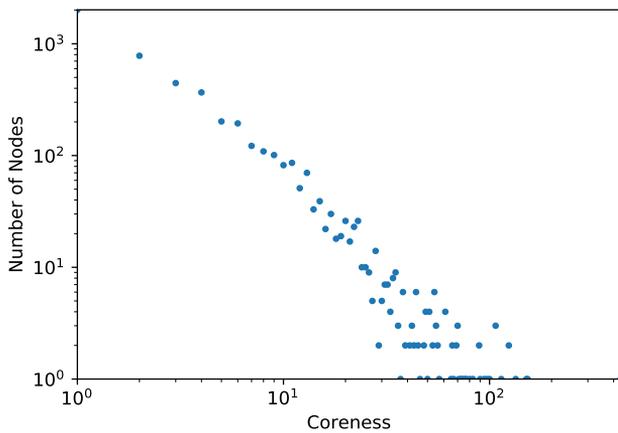
(a) Degree distribution



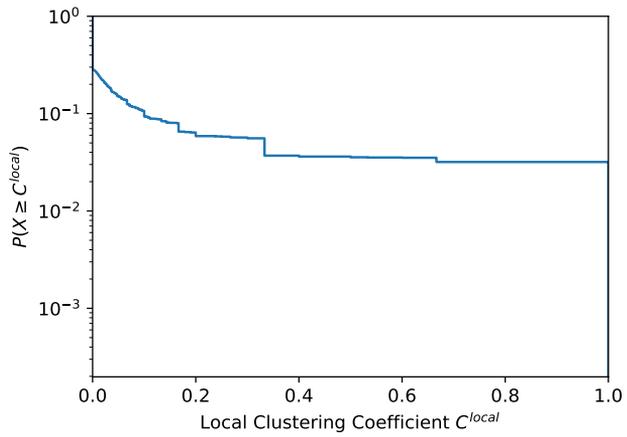
(b) Indegree distribution



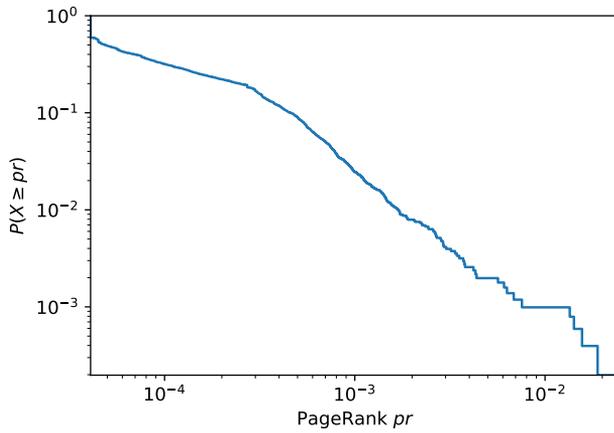
(c) Outdegree distribution



(d) Coreness distribution

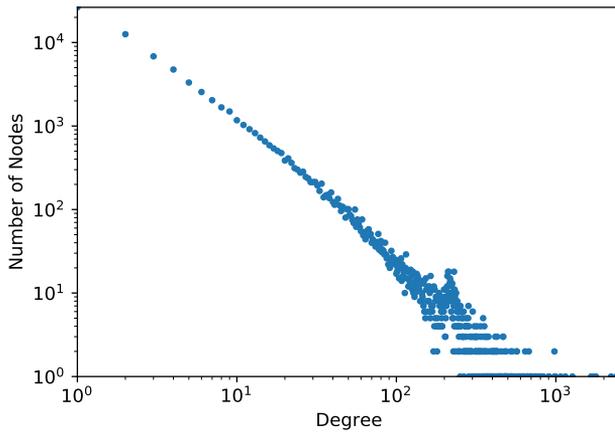


(e) Local clustering coefficient distribution

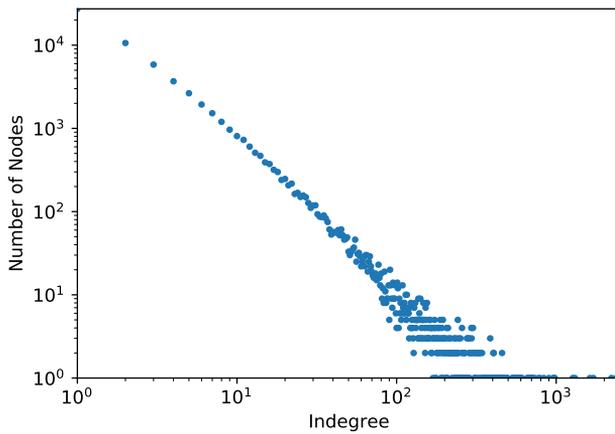


(f) PageRank distribution

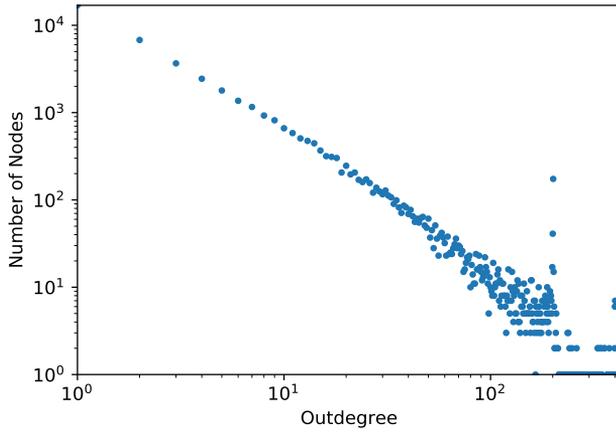
Figure B.24: The base feature distribution of the ARIS network.



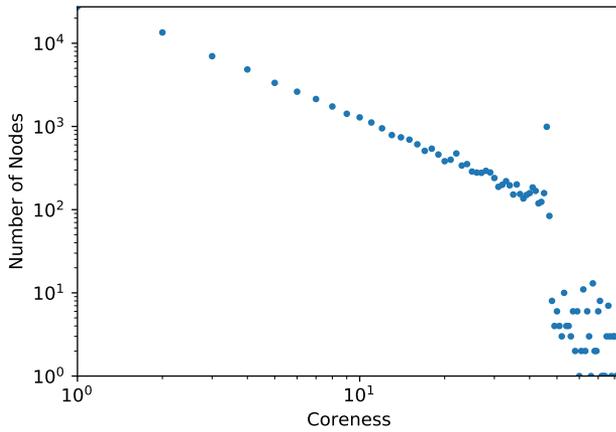
(a) Degree distribution



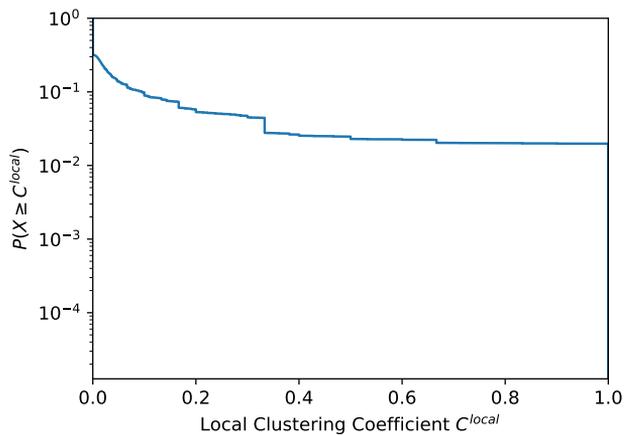
(b) Indegree distribution



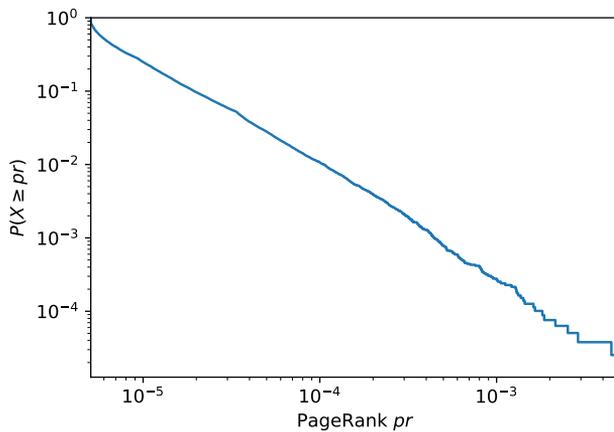
(c) Outdegree distribution



(d) Coreness distribution

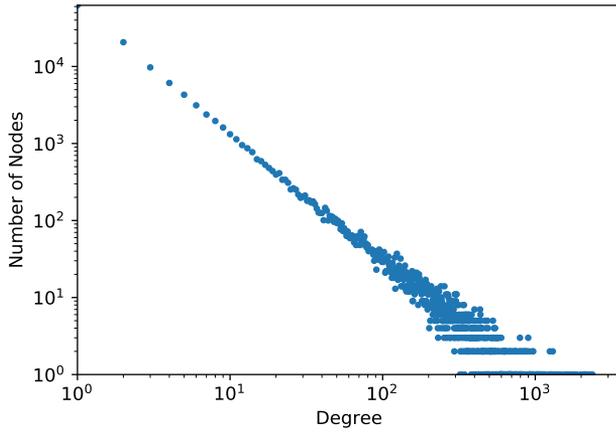


(e) Local clustering coefficient distribution

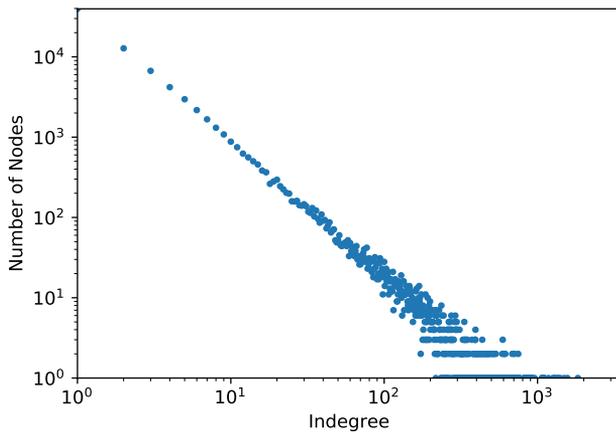


(f) PageRank distribution

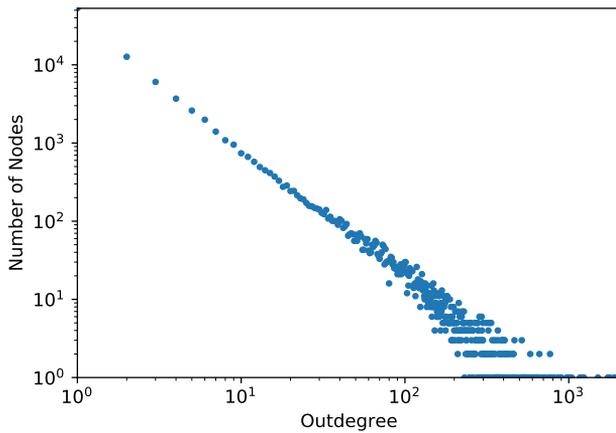
Figure B.25: The base feature distribution of the Slashdot-Zoo network.



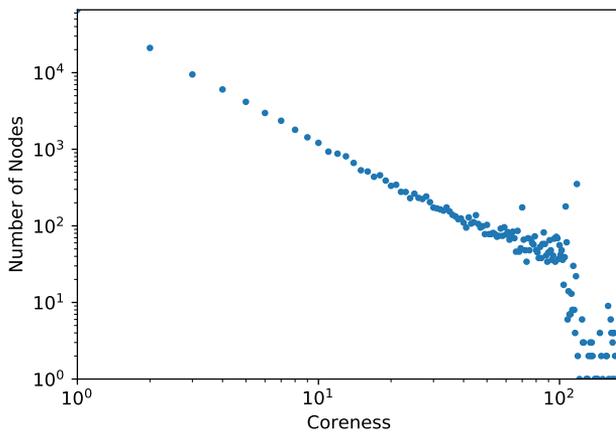
(a) Degree distribution



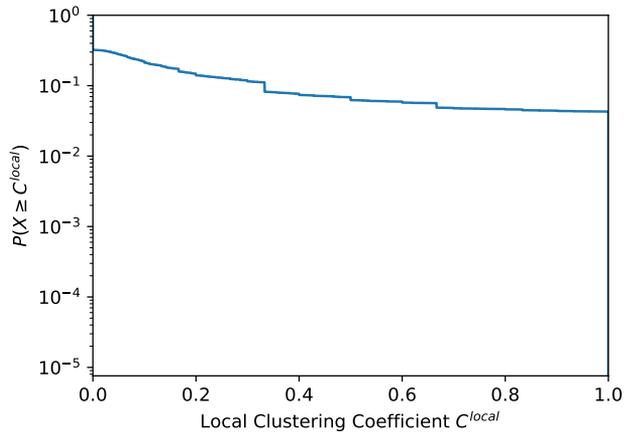
(b) Indegree distribution



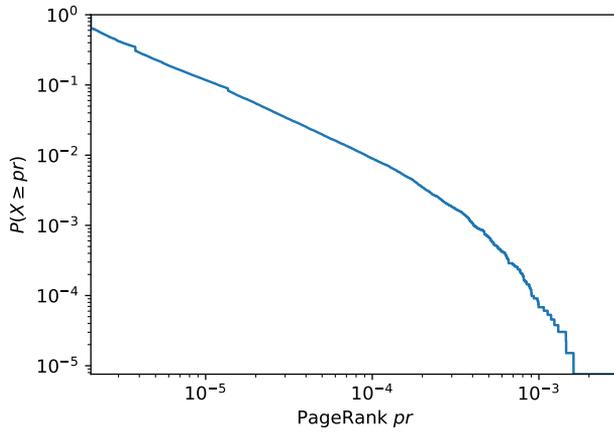
(c) Outdegree distribution



(d) Coreness distribution



(e) Local clustering coefficient distribution



(f) PageRank distribution

Figure B.26: The base feature distribution of the Epinion-Trust network.

APPENDIX: WIKI-TALK DATASET

In this appendix, we describe the Wiki-talk datasets, which consist of the user interaction networks of all user talk pages in Wikipedia, in 28 languages. Each user is represented by her original Wikipedia user ID, and is assigned a role, according to her access level in Wikipedia. We also show how to use the parser provided by us to keep the data up-to-date and how to customise the datasets.

C.1 DESCRIPTION

In Wikipedia, each registered user has a talk page that can be used for discussion. We extract the user interaction networks of all user talk pages of Wikipedia in the 28 languages with the highest number of articles (at the time of dataset creation). The 28 languages are, listed alphabetically by their ISO 639 code: ar, bn, br, ca, cy, de, el, en, eo, es, eu, fr, gl, ht, it, ja, lv, nds, nl, oc, pl, pt, ru, sk, sr, sv, vi, zh.

Each language forms an individual directed network, in which each node is a Wikipedia user represented by her original Wikipedia user ID, and each directed edge (User_ID_A, User_ID_B, timestamp) represents a user interaction: User A wrote a message on User B's talk page at a certain time.

C.1.1 *User roles*

In the dataset, each user has an access level [79] that is defined by Wikipedia, which we interpret as the following roles:

- **Administrator.** Administrators refer to the accounts that have high level of access to contents and maintenance tools in Wikipedia. We combine the users that are granted as “sysops” or “bureaucrat” by the communities at RfA or RfB³, and label them as administrators.

³ *Requests for adminship (RfA), Requests for bureaucratship (RfB)*

- **Bot.** Bots are used in Wikipedia for (semi-)automatically improving contents. Bot accounts are marked as “bot” by an administrator, and each has specific tasks that it performs [78].
- **Normal user.** We categorise other users as normal users, who are not categorised as administrators or bots.

C.2 INSIGHTS

Table 2 shows some basic statistics of the datasets. All sub-datasets are denoted as their language codes, e.g., *de* stands for the data from the German Wikipedia. As we can see, all 28 networks have a variety of sizes, from 504 (*ht*) to around 3 million (*en*). The proportions of both bots and administrators are very small, although they vary highly among all sub-datasets, from 0.0027% to 5.97% and from 0% to 0.72% respectively.

C.3 DOWNLOAD AND PARSING

We parsed the Wikipedia dump files (xml) to Wiki-talk networks at the end of 2015. Users can download the parsed datasets at Zenodo¹ which is an open platform for dataset sharing. We have also open sourced the parsing tool² which we wrote in Clojure, in case users want to re-parse the datasets, or customise them, such as adding new roles.

C.3.1 *Parsing with Stu*

Using *Stu*³ to parse the dataset is the most convenient way. The only file that is needed is *main.stu*. Simply type in *stu* or, preferably:

```
nohup stu -k -j 3 &
```

Stu will automatically start downloading this program and the dump files and parse them.

¹ <https://dx.doi.org/10.5281/zenodo.49561>

² <https://github.com/yfiua/wiki-talk-parser>

³ <https://github.com/kunegis/stu>

C.3.2 Parsing without Stu

Parsing the dataset without Stu is also viable. However, multiple steps need to be taken care of manually.

- **Installation**

Manually download the latest jar files from the “release” page. Users need to download the Wikipedia dump files in xml format from the Wikipedia website⁴, too.

- **Parse**

```
java -jar parser.jar *input-file* *lang* > *output-file*
```

- **Shrink**

“Shrink” the resulted network, so to make unweighted directed networks without loops, as in the SNAP datasets [43].

```
java -jar shrinker.jar *input-file* > *output-file*
```

- **Group users**

Group users according to their roles.

```
java -jar grouper.jar *input-file* > *output-file*
```

- **Compilation (optional)**

Compile the parsing tool.

```
lein with-profile parser:shrinker:grouper uberjar
```

C.4 LICENSE

The datasets are published under the Creative Commons Attribution Share-Alike (CC BY-SA) 4.0 License [68]. The parsing tool is distributed under the Eclipse Public License either version 1.0 or any later version.

⁴ <https://dumps.wikimedia.org>

Table 2: Meta information of Wiki-talk networks

Lang	# Nodes	# Edges	# Bots	# Admins	% Bots	% Admins
ar	1095799	1913103	30	37	0.0027%	0.0034%
bn	83803	122078	18	16	0.0215%	0.0191%
br	1181	13754	43	8	3.6410%	0.6774%
ca	79736	351610	179	25	0.2245%	0.0314%
cy	2233	10740	39	16	1.7465%	0.7165%
de	519403	6729794	328	246	0.0631%	0.0474%
el	40254	190279	58	20	0.1441%	0.0497%
en	2987535	24981163	278	1313	0.0093%	0.0439%
eo	7586	47070	130	21	1.7137%	0.2768%
es	497446	2702879	34	75	0.0068%	0.0151%
eu	40993	58120	81	10	0.1976%	0.0244%
fr	1420367	4641928	97	163	0.0068%	0.0115%
gl	8097	63809	12	14	0.1482%	0.1729%
ht	536	1530	32	0	5.9701%	0.0000%
it	863846	3067680	137	104	0.0159%	0.0120%
ja	397635	1031378	51	49	0.0128%	0.0123%
lv	41424	73900	57	11	0.1376%	0.0266%
nds	23132	27432	56	5	0.2421%	0.0216%
nl	225749	1554699	237	50	0.1050%	0.0221%
oc	3144	11059	51	4	1.6221%	0.1272%
pl	155820	1358426	55	115	0.0353%	0.0738%
pt	541355	2424962	205	64	0.0379%	0.0118%
ru	457017	2282055	77	90	0.0168%	0.0197%
sk	41452	131884	105	8	0.2533%	0.0193%
sr	103068	312837	132	22	0.1281%	0.0213%
sv	120833	598066	41	72	0.0339%	0.0596%
vi	338714	607087	123	23	0.0363%	0.0068%
zh	1219241	2284546	93	77	0.0076%	0.0063%

NOMENCLATURE

η	The fitness of a node
$\hat{f}, \mathcal{L}\{f\}$	The Laplace transform of function f
A	The adjacency matrix of a graph
D	The degree matrix of a graph
pr	The PageRank of a node
Π	The preferential attachment selection probability of a node
$\rho(\eta)$	The fitness distribution
τ	The age of a node in physical time
c^{core}	The coreness of a node
C^{local}	The local Clustering coefficient of a node
E	The set of edges (links) in a network
$f \star g$	The convolution of functions f and g
G	A graph or a network
$g(t)$	The derivative of the network size with respect to the physical time t
$h(\tau)$	The average degree increase of a node as a function of the node age τ
k	The degree of a node
k^{in}	The indegree of a node
k^{out}	The outdegree of a node

M	The volume of a network
N	The size of a network
$P(k)$	The degree distribution as probability mass function
$p(k)$	The degree distribution as probability density function
$R(\tau)$	The ageing function
s	The system time
s_i	The system time at which node v_i joins the network
t	The physical time
t_i	The physical time at which node v_i joins the network
V	The set of vertices (nodes) in a network
Z	The normalisation factor, a.k.a. the partition function

LIST OF FIGURES

Figure 2.1	The SimGraph network.	20
Figure 2.2	The theoretical degree distribution of the Erdős-Rényi model $G(5000, 0.3)$, and the degree distribution of an instance of a synthetic graph generated from the model.	24
Figure 2.3	The degree distribution of the Gowalla network. Panel (a) plots the number of nodes (frequency) against degree k ; Panel (b) shows the cumulative degree distribution (complementary cumulative distribution function).	26
Figure 3.1	The expected degree growth curves of three nodes that join the network at different times in the Barabási-Albert model, as functions of the (a) system time; (b) node age.	37
Figure 3.2	The degree distributions of two synthetic networks with $N = 50000$, (a) of the Barabási-Albert model; (b) where preferential attachment is absent, and the selection probability Π is the same for all nodes despite of their degrees. Note (a) uses the log-log scale, while (b) uses the linear-log scale.	38
Figure 3.3	The expected degree growth curves of four nodes with different fitness and birth time in the Bianconi-Barabási model. The slopes of the growth curves depend solely on the fitness values, allowing the degrees of nodes with later birth time to surpass earlier ones. However, the strong first-mover-advantage is still present, because the degrees of nodes with the same fitness values still strongly depend on their birth time. Note this plot uses the log-log scale.	41
Figure 3.4	Illustration of the energy diagrams corresponding to a Bose gas (a) under high temperature; (b) in the Bose-Einstein condensation phase. Each energy level is associated with an energy ϵ . In (a) the bosons are distributed over a wide range of energy levels. In (b) almost all bosons land in the lowest energy level, while only few bosons populate the higher energy levels sparsely.	43
Figure 3.5	The degree distribution of an instance of a synthetic graph generated from the superlinear preferential attachment model $\Pi \sim k^{1.2}$ with 5×10^4 nodes. The green line indicates the curve of the theoretical degree distribution of the Barabási-Albert model with linear preferential attachment.	47
Figure 3.6	The average number of citations as a function of paper age for papers published in different time periods in (a) the APS data and (b) the DBLP data. Papers are grouped by their publication year. Note that the plots use the log-log scale.	50

Figure 3.7 The network growth curve of (a) the APS data and (b) the DBLP data. Each plot shows how the total number of papers grows with time in each dataset. Note that the plots use the linear-log scale. 52

Figure 3.8 The average outdegree of papers published in different years in (A) the APS data and (B) the DBLP data. 54

Figure 3.9 The average number of citations as a function of the paper age, rescaled by the average outdegree in the years when the citations were received. Papers are divided in three groups by their publication year. 55

Figure 3.10 The indegree distributions of (a) the APS data and (b) the DBLP data. Estimated power-law exponents (for $k > 5$) are: 1.906 for APS and 1.947 for DBLP. 57

Figure 3.11 The expected degree growth curves of three nodes which join the network at different times (with the same fitness value) in the Bianconi-Barabási model. 58

Figure 3.12 The theoretical degree growth curves under the exponential ageing $R(\tau) = e^{-\gamma\tau}$ with $\gamma = 0.5$ and $\theta_{\text{exp}} = 0.33$ as in Equation (3.57). The dashed lines show the corresponding theoretical asymptotes as in Equation (3.59). Note the X axis is extended to the negative domain in order to better illustrate the sigmoid shapes. 65

Figure 3.13 Subfigure (a) shows the fitness distribution $\rho(\eta) = (\lambda + 1)(1 - \eta)^\lambda$ with different three parameters λ . When λ is larger, the fitness distribution is more skewed. Subfigure (b) shows the corresponding σ as a function of λ solved using a numerical solver, which fails to find solutions when $\lambda > 1$ 82

Figure 3.14 The network growth curve (red) of a synthetic network generated from the linear preferential attachment ($dk/d\tau \sim k$ as in the Barabási-Albert model) using our framework. The slope of the green dashed line indicates the theoretical estimation of the asymptotic network growth. The blue dashed curve shows the growth of the maximum degree in the network. Note that the plot uses the linear-log scale to illustrate the exponential curves. 88

Figure 3.15 The degree distribution of a synthetic network generated from the linear preferential attachment ($dk/d\tau \sim k$ as in the Barabási-Albert model) using our framework. The green curves indicate the theoretical estimation of the degree distributions. Note that the plot uses the log-log scale to illustrate the scale-free distributions. 89

Figure 3.16 The average degree growth curves of nodes with different ages in a synthetic network generated from the linear preferential attachment ($dk/d\tau \sim k$ as in the Barabási-Albert model) using our framework. The slope of the red dashed line indicates the theoretical estimation of the degree growth. Note that the plot uses the linear-log scale to illustrate the exponential curves. 90

Figure 3.17	The network growth curves (red) of synthetic networks generated using our framework with preferential attachment with heterogeneous fitness ($\Pi \sim k\eta$ as in the Bianconi-Barabási model). The green dashed lines indicate the asymptotic slopes of the theoretical estimation of the network growth. The blue dashed curves show the growth of the maximum degree in the networks. Note that the plots use the linear-log scale.	92
Figure 3.18	The degree distributions of the synthetic networks generated using our framework with preferential attachment with heterogeneous fitness ($\Pi \sim k\eta$ as in the Bianconi-Barabási model). The green lines show the theoretical estimation. Note that the plots use the log-log scale.	93
Figure 3.19	The average degree growth curves of four groups of nodes with different time when joining the network and different fitness values. The synthetic network is generated using our framework with preferential attachment with heterogeneous fitness ($\Pi \sim k\eta$ as in the Bianconi-Barabási model). The dashed lines show the theoretical estimation. Note that the plot uses the linear-log scale.	94
Figure 3.20	The network growth curves (red) of synthetic networks generated using our framework with preferential attachment with heterogeneous fitness and (a) ageing as power function; (b) ageing as exponential function. ($\Pi \sim k\eta R(\tau)$ as in the relevance decay model). The green dashed lines indicate the asymptotic slopes of the theoretical estimation of the network growth. The blue dashed curves show the growth of the maximum degree in the networks. Note that the plots use the linear-log scale.	96
Figure 3.21	The degree distributions of the synthetic networks generated using our framework with preferential attachment with heterogeneous fitness and (a) ageing as power function; (b) ageing as exponential function. ($\Pi \sim k\eta R(\tau)$ as in the relevance decay model). The green lines show the theoretical estimation. Note that the plots use the log-log scale.	97
Figure 3.22	The average degree growth curves of groups of nodes with approximately the same age and different fitness values. The synthetic network is generated using our framework with preferential attachment with heterogeneous fitness and (a) ageing as power function; (b) ageing as exponential function. ($\Pi \sim k\eta R(\tau)$ as in the relevance decay model). The dashed lines show the theoretical estimation. Note that (a) uses the log-log scale, while (b) uses the linear-linear scale.	98

Figure 3.23 The network growth curve (red) of a synthetic network generated from the superlinear preferential attachment ($dk/d\tau \sim k^{1.2}$) using our framework. The green dashed line indicates the theoretical estimation of the asymptotic network growth, where the network size should go to infinity at $t = 5$ (indicated with the vertical grey dashed line). The blue dashed curve shows the growth of the maximum degree in the network. Note that the plot uses the linear-log scale. 99

Figure 3.24 The degree distribution of a synthetic network generated from the superlinear preferential attachment ($dk/d\tau \sim k^{1.2}$) using our framework. Note that the plot uses the log-log scale. . . . 100

Figure 3.25 The average degree growth curve of a group of 10 nodes which joined the network at approximately the same time $t = 1$. The synthetic network is generated from the superlinear preferential attachment ($dk/d\tau \sim k^{1.2}$) using our framework. The red dashed line indicates the theoretical estimation of the degree growth. Note that the plot uses the linear-log scale. 100

Figure 4.1 Overview of TraNet, the transfer learning approach for user study described in this chapter. We learn from the source dataset how to detect the following three (exemplary) pre-defined user labels: central users (diamonds), bridging users (triangles), and normal users (circles); and transfer the knowledge to the unlabelled target network to predict the user labels in it. The components within the source domain D_S and within the common domain D' but above the dashed line are obtained during *training*, while the ones within the target domain D_T and within the common domain D' but below the dashed line are obtained during *application*. 110

Figure 4.2 The base feature distribution of the Wiki-talk-en network. . . . 116

Figure 4.3 Degree distributions of networks Wiki-talk-de and Wiki-talk-fr, before and after the power-law transformation. Each dot in the plot represents the probability (Y axis) of a degree value (X axis) in the network. Two separated curves overlap after the transformation. 119

Figure 4.4 Convergence procedure of aggregating neighbourhood features for (a) degree, (b) local clustering coefficient, (c) PageRank and (d) coreness in different datasets (each curve represents a dataset). The X axis shows the round number of feature aggregation, while the Y axis shows the maximum absolute value of the Pearson correlation coefficient ρ between the newly generated neighbourhood features of the current round and the previous round. New features with bigger ρ provide less information [29]. In most cases, ρ gets larger than 0.9 (indicated by the dashed horizontal lines) within 5 rounds. We omit other datasets and features here, since they show similar patterns. . . 123

Figure 4.5 ROC-AUC performance of the classifiers using different settings (see Section 4.3.2). In each box plot, one data point corresponds to the ROC-AUC score of a classifier that transfers user roles from one Wiki-talk network to another. The orange bar shows the median value, while the green triangle shows the mean value of the ROC-AUC in each experiment. Our approach TraNet achieves the best average ROC-AUC among all transfer learning approaches in both classification tasks. Note TrAdaBoost is omitted here due to its poor performance. 129

Figure A.1 The network growth curves of synthetic networks generated using our framework with preferential attachment with heterogeneous fitness ($\Pi \sim k\eta$ as in the Bianconi-Barabási model). In each plot the parameter λ of the fitness distribution $\rho(\eta) \sim (1 - \eta)^\lambda$ is set different. 139

Figure A.2 The network growth curves of the synthetic networks generated using our framework with preferential attachment with heterogeneous fitness and power ageing ($\Pi \sim k\eta R(\tau)$). In each plot, the parameters λ of the fitness distribution $\rho(\eta) \sim (1 - \eta)^\lambda$, and γ of the power ageing function $R(\tau) = (\tau + 1)^{-\gamma}$, are set different. 142

Figure A.3 The network growth curves of the synthetic networks generated using our framework with preferential attachment with heterogeneous fitness and exponential ageing ($\Pi \sim k\eta R(\tau)$). In each plot, the parameters λ of the fitness distribution $\rho(\eta) \sim (1 - \eta)^\lambda$, and γ of the exponential ageing function $R(\tau) = e^{-\gamma\tau}$, are set different. 144

Figure A.4 The degree distributions of the synthetic networks generated using our framework with preferential attachment with heterogeneous fitness ($\Pi \sim k\eta$ as in the Bianconi-Barabási model). In each plot the parameter λ of the fitness distribution $\rho(\eta) \sim (1 - \eta)^\lambda$ is set different. 151

Figure A.5 The degree distributions of the synthetic networks generated using our framework with preferential attachment with heterogeneous fitness and power ageing ($\Pi \sim k\eta R(\tau)$). In each plot, the parameters λ of the fitness distribution $\rho(\eta) \sim (1 - \eta)^\lambda$, and γ of the power ageing function $R(\tau) = (\tau + 1)^{-\gamma}$, are set different. 154

Figure A.6 The degree distributions of the synthetic networks generated using our framework with preferential attachment with heterogeneous fitness and exponential ageing ($\Pi \sim k\eta R(\tau)$). In each plot, the parameters λ of the fitness distribution $\rho(\eta) \sim (1 - \eta)^\lambda$, and γ of the exponential ageing function $R(\tau) = e^{-\gamma\tau}$, are set different. 156

Figure B.1 The base feature distribution of the Wiki-talk-ar network. 160

Figure B.2 The base feature distribution of the Wiki-talk-ca network. 163

Figure B.3 The base feature distribution of the Wiki-talk-de network. 166

Figure B.4 The base feature distribution of the Wiki-talk-el network. 169

Figure B.5	The base feature distribution of the Wiki-talk-en network. . . .	172
Figure B.6	The base feature distribution of the Wiki-talk-eo network. . . .	175
Figure B.7	The base feature distribution of the Wiki-talk-es network. . . .	178
Figure B.8	The base feature distribution of the Wiki-talk-eu network. . . .	181
Figure B.9	The base feature distribution of the Wiki-talk-fr network. . . .	184
Figure B.10	The base feature distribution of the Wiki-talk-it network. . . .	187
Figure B.11	The base feature distribution of the Wiki-talk-ja network. . . .	190
Figure B.12	The base feature distribution of the Wiki-talk-lv network. . . .	193
Figure B.13	The base feature distribution of the Wiki-talk-nds network. . . .	196
Figure B.14	The base feature distribution of the Wiki-talk-nl network. . . .	199
Figure B.15	The base feature distribution of the Wiki-talk-oc network. . . .	202
Figure B.16	The base feature distribution of the Wiki-talk-pl network. . . .	205
Figure B.17	The base feature distribution of the Wiki-talk-pt network. . . .	208
Figure B.18	The base feature distribution of the Wiki-talk-ru network. . . .	211
Figure B.19	The base feature distribution of the Wiki-talk-sk network. . . .	214
Figure B.20	The base feature distribution of the Wiki-talk-sr network. . . .	217
Figure B.21	The base feature distribution of the Wiki-talk-sv network. . . .	220
Figure B.22	The base feature distribution of the Wiki-talk-vi network. . . .	223
Figure B.23	The base feature distribution of the Wiki-talk-zh network. . . .	226
Figure B.24	The base feature distribution of the ARIS network.	230
Figure B.25	The base feature distribution of the Slashdot-Zoo network. . . .	233
Figure B.26	The base feature distribution of the Epinion-Trust network. . .	236

LIST OF TABLES

Table 1	Predicting trusted users in the target network ARIS with the knowledge transferred from the two source networks Slashdot-Zoo and Epinion-Trust respectively. The values in the table show the ROC-AUC performance of the classifier in different settings (see Section 4.3.2). We can achieve the best performance with transfer learning using our approach TraNet.	130
Table 2	Meta information of Wiki-talk networks	240

BIBLIOGRAPHY

- [1] Lada A Adamic and Bernardo A Huberman. "Power-law distribution of the World Wide Web." In: *Science* 287.5461 (2000), pp. 2115–2115.
- [2] Eneko Agirre and Oier Lopez De Lacalle. "On robustness and domain adaptation using SVD for word sense disambiguation." In: *Proceedings of the 22nd International Conference on Computational Linguistics*. Vol. 1. 2008, pp. 17–24.
- [3] Réka Albert and Albert-László Barabási. "Statistical mechanics of complex networks." In: *Reviews of Modern Physics* 74.1 (2002), p. 47.
- [4] Andrew Arnold, Ramesh Nallapati, and William W Cohen. "A comparative study of methods for transductive transfer learning." In: *International Conference on Data Mining Workshops*. 2007, pp. 77–82.
- [5] Albert-László Barabási and Réka Albert. "Emergence of scaling in random networks." In: *Science* 286.5439 (1999), pp. 509–512.
- [6] Albert-László Barabási, Réka Albert, and Hawoong Jeong. "Mean-field theory for scale-free random networks." In: *Physica A: Statistical Mechanics and its Applications* 272.1-2 (1999), pp. 173–187.
- [7] Albert-László Barabási et al. *Network Science*. Cambridge University Press, 2016.
- [8] Klaus Berberich, Srikanta Bedathur, Gerhard Weikum, and Michalis Vazirgiannis. "Comparing apples and oranges: normalized PageRank for evolving graphs." In: *International Conference on World Wide Web*. 2007, pp. 1145–1146.
- [9] Ginestra Bianconi and Albert-László Barabási. "Bose-Einstein condensation in complex networks." In: *Physical Review Letters* 86.24 (2001), p. 5632.
- [10] Ginestra Bianconi and Albert-László Barabási. "Competition and multiscaling in evolving networks." In: *EPL (Europhysics Letters)* 54.4 (2001), p. 436.
- [11] Elisabetta Binaghi and Anna Rampini. "Fuzzy decision making in the classification of multisource remote sensing data." In: *Optical Engineering* 32.6 (1993), pp. 1193–1204.
- [12] Béla Bollobás, Oliver Riordan, Joel Spencer, Gábor Tusnády, et al. "The degree sequence of a scale-free random graph process." In: *Random Structures & Algorithms* 18.3 (2001), pp. 279–290.
- [13] John Adrian Bondy, Uppaluri Siva Ramachandra Murty, et al. *Graph theory with applications*. Vol. 290. Macmillan London, 1976.
- [14] Anna D Broido and Aaron Clauset. "Scale-free networks are rare." In: *Nature communications* 10.1 (2019), pp. 1–10.
- [15] Guido Caldarelli, Andrea Capocci, Paolo De Los Rios, and Miguel A Munoz. "Scale-free networks from varying vertex intrinsic fitness." In: *Physical Review Letters* 89.25 (2002), p. 258702.

- [16] Andrea Capocci, Vito DP Servedio, Francesca Colaiori, Luciana S Buriol, Debora Donato, Stefano Leonardi, and Guido Caldarelli. "Preferential attachment in the growth of social networks: The internet encyclopedia Wikipedia." In: *Physical Review E* 74.3 (2006), p. 036116.
- [17] Eunjoon Cho, Seth A Myers, and Jure Leskovec. "Friendship and mobility: user movement in location-based social networks." In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2011, pp. 1082–1090.
- [18] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. "Power-law distributions in empirical data." In: *SIAM Review* 51.4 (2009), pp. 661–703.
- [19] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. "Boosting for transfer learning." In: *Proceedings of the 24th International Conference on Machine Learning*. 2007, pp. 193–200.
- [20] Munmun De Choudhury, Yu-Ru Lin, Hari Sundaram, Kasim Selcuk Candan, Lexing Xie, and Aisling Kelliher. "How does the data sampling strategy impact the discovery of information diffusion in social media?" In: *Fourth International AAAI Conference on Weblogs and Social Media*. 2010.
- [21] Sergey N Dorogovtsev and José Fernando F Mendes. "Effect of the accelerating growth of communications networks on their structure." In: *Physical Review E* 63.2 (2001), p. 025101.
- [22] Sergey N Dorogovtsev, José Fernando F Mendes, and Alexander N Samukhin. "Structure of growing networks with preferential linking." In: *Physical Review Letters* 85.21 (2000), p. 4633.
- [23] Nathan Eagle and Alex Sandy Pentland. "Reality mining: sensing complex social systems." In: *Personal and ubiquitous computing* 10.4 (2006), pp. 255–268.
- [24] Albert Einstein. "Quantentheorie des einatomigen idealen Gases." In: *Albert Einstein: Akademie-Vorträge: Sitzungsberichte der Preussischen Akademie der Wissenschaften 1914–1932* (2005), pp. 237–244.
- [25] Paul Erdős and Alfréd Rényi. "On random graphs." In: *Publicationes Mathematicae Debrecen* 6.26 (1959), pp. 290–297.
- [26] Tom Fawcett. "An introduction to ROC analysis." In: *Pattern recognition letters* 27.8 (2006), pp. 861–874.
- [27] Michael J Gagen and JS Mattick. "Accelerating, hyperaccelerating, and decelerating networks." In: *Physical Review E* 72.1 (2005), p. 016123.
- [28] RP Grimaldi. *Discrete and Combinatorial Mathematics: An Applied Introduction*, ser. Addison-Wesley World Student Series. Addison-Wesley, 1994.
- [29] Mark A. Hall. "Correlation-based Feature Selection for Machine Learning." PhD thesis. The University of Waikato, 1999.
- [30] Douglas M Hawkins. "The problem of overfitting." In: *Journal of Information and Computer Sciences* 44.1 (2004), pp. 1–12.

- [31] Keith Henderson, Brian Gallagher, Tina Eliassi-Rad, Hanghang Tong, Sugato Basu, Leman Akoglu, Danai Koutra, Christos Faloutsos, and Lei Li. "RoIX: structural role extraction & mining in large graphs." In: *International Conference on Knowledge Discovery and Data Mining*. 2012, pp. 1231–1239.
- [32] Keith Henderson, Brian Gallagher, Lei Li, Leman Akoglu, Tina Eliassi-Rad, Hanghang Tong, and Christos Faloutsos. "It's who you know: Graph mining using recursive structural features." In: *International Conference on Knowledge Discovery and Data Mining*. 2011, pp. 663–671.
- [33] Joao P Hespanha. *Linear Systems Theory*. Princeton University Press, 2018.
- [34] Kerson Huang. *Statistical mechanics*. Wiley, 1987, p. 512.
- [35] Hawoong Jeong, Zoltan Néda, and Albert-László Barabási. "Measuring preferential attachment in evolving networks." In: *EPL (Europhysics Letters)* 61.4 (2003), p. 567.
- [36] Sepandar D Kamvar, Mario T Schlosser, and Hector Garcia-Molina. "The eigen-trust algorithm for reputation management in p2p networks." In: *Proceedings of the 12th International Conference on World Wide Web*. 2003, pp. 640–651.
- [37] Maksim Kitsak, Lazaros K Gallos, Shlomo Havlin, Fredrik Liljeros, Lev Muchnik, H Eugene Stanley, and Hernán A Makse. "Identification of influential spreaders in complex networks." In: *Nature physics* 6.11 (2010), pp. 888–893.
- [38] Jon M Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew S Tomkins. "The web as a graph: measurements, models, and methods." In: *International Computing and Combinatorics Conference*. Springer. 1999, pp. 1–17.
- [39] Dirk Koschützki, Katharina Anna Lehmann, Dagmar Tenfelde-Podehl, and Oliver Zlotowski. "Advanced centrality concepts." In: *Network Analysis*. 2005, pp. 83–111.
- [40] Paul L Krapivsky, Sidney Redner, and Francois Leyvraz. "Connectivity of growing random networks." In: *Physical Review Letters* 85.21 (2000), p. 4629.
- [41] Jérôme Kunegis, Andreas Lommatzsch, and Christian Bauckhage. "The Slashdot Zoo: Mining a Social Network with Negative Edges." In: *Proceedings of the International World Wide Web Conference*. 2009, pp. 741–750.
- [42] Daniel D Lee and H Sebastian Seung. "Algorithms for non-negative matrix factorization." In: *Advances in neural information processing systems*. 2001, pp. 556–562.
- [43] Jure Leskovec and Andrej Krevl. *SNAP Datasets: Stanford Large Network Dataset Collection*. 2014.
- [44] Michael Ley. "The DBLP computer science bibliography: Evolution, research issues, perspectives." In: *String Processing and Information Retrieval*. Springer. 2002, pp. 1–10.
- [45] Andy Liaw and Matthew Wiener. "Classification and regression by randomForest." In: *R news* 2.3 (2002), pp. 18–22.
- [46] Fritz London. "The λ -phenomenon of liquid helium and the Bose-Einstein degeneracy." In: *Nature* 141.3571 (1938), pp. 643–644.

- [47] Fragkiskos D. Malliaros, Apostolos N. Papadopoulos, and Michalis Vazirgiannis. "Core Decomposition in Graphs: Concepts, Algorithms and Applications." In: *Proceedings of the 19th International Conference on Extending Database Technology*. 2016, pp. 720–721.
- [48] Paolo Massa and Paolo Avesani. "Controversial Users Demand Local Trust Metrics: an Experimental Study on epinions.com Community." In: *Proceedings of the American Association for Artificial Intelligence Conference*. 2005, pp. 121–126.
- [49] Matúš Medo. "Statistical validation of high-dimensional models of growing networks." In: *Physical Review E* 89.3 (2014), p. 032801.
- [50] Matúš Medo, Giulio Cimini, and Stanislao Gualdi. "Temporal effects in the growth of networks." In: *Physical Review Letters* 107.23 (2011), p. 238701.
- [51] RV Mises and Hilda Pollaczek-Geiringer. "Praktische Verfahren der Gleichungsauflösung." In: *Zeitschrift für Angewandte Mathematik und Mechanik* 9.2 (1929), pp. 152–164.
- [52] Mark EJ Newman. "Clustering and preferential attachment in growing networks." In: *Physical Review E* 64.2 (2001), p. 025102.
- [53] Mark EJ Newman. "The first-mover advantage in scientific publication." In: *EPL (Europhysics Letters)* 86.6 (2009), p. 68001.
- [54] Mark EJ Newman. *Networks: An Introduction*. Oxford University Press, 2010.
- [55] Mark EJ Newman. *Networks*. Oxford University Press, 2018.
- [56] M. P. O'Brien and B. D. Sullivan. "Locally estimating core numbers." In: *IEEE International Conference on Data Mining* (2014), pp. 460–469.
- [57] FWJ Olver. "The generalized exponential integral." In: *Approximation and Computation: A Festschrift in Honor of Walter Gautschi*. Springer, 1994, pp. 497–510.
- [58] Thais Mayumi Oshiro, Pedro Santoro Perez, and José Augusto Baranauskas. "How many trees in a random forest?" In: *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. 2012, pp. 154–168.
- [59] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. *The PageRank citation ranking: Bringing order to the Web*. Tech. rep. 1999.
- [60] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. "Domain adaptation via transfer component analysis." In: *IEEE Transactions on Neural Networks* 22.2 (2011), pp. 199–210.
- [61] Sinno Jialin Pan and Qiang Yang. "A survey on transfer learning." In: *IEEE Transactions on knowledge and data engineering* 22.10 (2010), pp. 1345–1359.
- [62] Pietro Della Briotta Parolo, Raj Kumar Pan, Rumi Ghosh, Bernardo A Huberman, Kimmo Kaski, and Santo Fortunato. "Attention decay in science." In: *Journal of Informetrics* 9.4 (2015), pp. 734–745.
- [63] Garry Robins. "A tutorial on methods for the modeling and analysis of social network data." In: *Journal of Mathematical Psychology* 57.6 (2013), pp. 261–274.

- [64] Matthew Rowe, Miriam Fernandez, Harith Alani, Inbal Ronen, Conor Hayes, and Marcel Karnstedt. "Behaviour analysis across different types of Enterprise Online Communities." In: *Proceedings of the 4th Annual ACM Web Science Conference*. 2012, pp. 255–264.
- [65] Robert J Serfling. *Approximation theorems of mathematical statistics*. Vol. 162. 2009.
- [66] William C Stwalley and LH Nosanow. "Possible 'new' quantum systems." In: *Physical Review Letters* 36.15 (1976), p. 910.
- [67] Bongwon Suh, Gregorio Convertino, Ed H Chi, and Peter Pirolli. "The singularity is not near: Slowing growth of Wikipedia." In: *Proceedings of the 5th International Symposium on Wikis and Open Collaboration*. 2009, pp. 1–10.
- [68] Jun Sun and Jérôme Kunegis. *Wiki-talk Datasets*. Apr. 2016. doi: 10.5281/zenodo.49561. URL: <https://west.uni-koblenz.de/sites/default/files/publications/wiki-talk-description.pdf>.
- [69] Jun Sun, Jérôme Kunegis, and Steffen Staab. "Predicting User Roles in Social Networks using Transfer Learning with Feature Transformation." In: *Proceedings of the IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. 2016, pp. 128–135.
- [70] Jun Sun, Matúš Medo, and Steffen Staab. "Time-invariant degree growth in preferential attachment network models." In: *Physical Review E* 101 (2 2020), p. 022309.
- [71] Jun Sun, Steffen Staab, and Fariba Karimi. "Decay of Relevance in Exponentially Growing Networks." In: *Proceedings of the 10th ACM Conference on Web Science*. 2018, pp. 343–351.
- [72] Jun Sun, Steffen Staab, and Jérôme Kunegis. "Understanding social networks using transfer learning." In: *Computer* 51.6 (2018), pp. 52–60.
- [73] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. "Arnet-Miner: Extraction and Mining of Academic Social Networks." In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2008, pp. 990–998.
- [74] Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P Gummadi. "On the evolution of user interaction in facebook." In: *Proceedings of the 2nd ACM workshop on Online social networks*. 2009, pp. 37–42.
- [75] Dashun Wang, Chaoming Song, and Albert-László Barabási. "Quantifying Long-Term Scientific Impact." In: *Science* 342.6154 (2013), pp. 127–132. ISSN: 0036-8075.
- [76] Stanley Wasserman, Katherine Faust, et al. *Social network analysis: Methods and applications*. Vol. 8. Cambridge university press, 1994.
- [77] Duncan J Watts and Steven H Strogatz. "Collective dynamics of 'small-world' networks." In: *Nature* 393.6684 (1998), pp. 440–442.
- [78] Wikipedia. *Bot policy* — *Wikipedia, The Free Encyclopedia*. [Accessed 25-July-2016]. 2016. URL: https://en.wikipedia.org/w/index.php?title=Wikipedia:Bot_policy&oldid=730456542.

- [79] Wikipedia. *User access levels* — *Wikipedia, The Free Encyclopedia*. [Accessed 13-July-2016]. 2016. URL: https://en.wikipedia.org/w/index.php?title=Wikipedia:User_access_levels&oldid=727508916.
- [80] Xindong Wu, Vipin Kumar, J Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J McLachlan, Angus Ng, Bing Liu, S Yu Philip, et al. "Top 10 algorithms in data mining." In: *Knowledge and Information Systems* 14.1 (2008), pp. 1-37.

DECLARATION

Erklärung über die Eigenständigkeit der Dissertation

Ich versichere, dass ich die vorliegende Arbeit mit dem Titel „Phenomena in growing networks and learning across networks“ selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe; aus fremden Quellen entnommene Passagen und Gedanken sind als solche kenntlich gemacht.

Declaration of authorship

I hereby certify that the dissertation entitled “Phenomena in growing networks and learning across networks” is entirely my own work except where otherwise indicated. Passages and ideas from other sources have been clearly indicated.

Koblenz, May 9, 2021

Jun Sun