Homophily Rearrangement Algorithms and Similarity Based Diffusion on Networks

Benedek András Rózemberczki

Central European University
Department of Economics

In partial fulfillment of the requirements for the degree of Master of Arts

Supervisor: Professor Rosario Nunzio Mantegna

Budapest, Hungary
2016

Abstract

The thesis first proposes three novel algorithms that can rearrange generic vertex features of a network in a way that target levels of homophily or heterophily regarding the variables are achievable. The thesis generalizes these algorithms to multivariate networks where vertices have more than a single generic vertex feature. Simulation results demonstrate that the homophily rearrangement algorithms' expected convergence time to a target homophily level depends on the network size, the target homophily level and in multivariate cases on the correlation among the generic vertex features. In addition, I extend the susceptible-infect model in a way that the transmission probability between vertices depends on the dissimilarity of generic vertex features. With the ability to control the level of homophily on the network with homophily rearrangement networks, the properties of the proposed similarity based diffusion model can be investigated. Empirical results establish the phenomenon that homophily propagates similarity based diffusion. Moreover, I demonstrate that the opposite is true when agents discriminate based on breed and the initial seeding originates from the discriminating breed.

JEL Classification: C12, C63, D85

Keywords: Agent-Based Modeling, Diffusion, Homophily, Susceptible-Infected Model

Acknowledgments

I wish to express my heartfelt gratitude towards my supervisor Rosario Nunzio Mantegna. His critiques and recommendations guided me during research and the thesis writing process. Furthermore, I am indebted to Zsuzsanna Tóth, from the Center for Academic Writing, who provided excellent advices regarding the style and structure of my thesis. The R sessions of Luca Marotta helped my development with respect to statistical programming skills which allowed for the implementation of fairly complex algorithms and experiments in my thesis. Finally, I am obliged to Enikő and my family for their understanding and acquittance.

Contents

List of Figures

List of Tables

List of Algorithms

List of Scripts

Chapter 1: Introduction

The thesis in hand focuses on three closely related research questions about homophily on complex networks. Algorithms and models that are introduced in order to answer the proposed research questions help to deepen our current understanding of homophily and similarity based spreading processes on networks. A major goal of my research is to investigate homophily and diffusion in a theoretically unified modeling framework. In the next three paragraphs I am going to elaborate on the proposed research questions and argue about their connectedness and relevance.

First, my thesis investigates whether it is possible to construct algorithms that can generate homophilous networks (regarding a single feature) without changing the network topology and the investigated feature's distribution. This question is relevant, because different assignments of a generic vertex property (which has a preset empirical distribution) might result in the same level of homophily on the network. From this it comes that the feature vector's correlation with the topological vertex properties might be heterogeneous. These heterogeneous correlations imply that vertices that have the same feature value have different functional roles in networks while the macro-level similarity of neighbors is the same in the system. In addition, it will be shown, that the existing network topology is a constraint which is a possible obstacle to the potential level of homophily on the network.

Second, real networks exhibit homophily towards multiple generic features, it is crucial to invent methods that generate homophilous networks with respect to multiple features. The thesis raises the question, whether the proposed univariate algorithms are extendable in order to deal with systems that have multiple generic vertex features. This question is important as generic vertex features that are behind homophily are possibly correlated with each other and identifying the true factors behind homophily is essential when one considers the true cause of homophily. Generating networks that show homophily

regarding multiple variables while the features' distribution is fixed might help our understanding of the endogenous homophily phenomenon.

Third, the similarity of nodes regarding generic properties might influence the diffusion of information, technologies, diseases or beliefs on the investigated network. Moreover, the homophily in the network is an emergent phenomenon which has roots in local similarity. These two statements together imply that there might be a connection between diffusion and homophily. The algorithms that answer the first and second research questions are able to control homophily. With the ability to control homophily one is able to investigate how homophily affects the diffusion on the network. Current research on this simple phenomenon, namely the connection between homophily and diffusion, uses modeling approaches that are different from the ones that I will use in my thesis. To put it simply, I propose a modification of the susceptible infected model to investigate properties of similarity based diffusion on networks where the homophily is controlled.

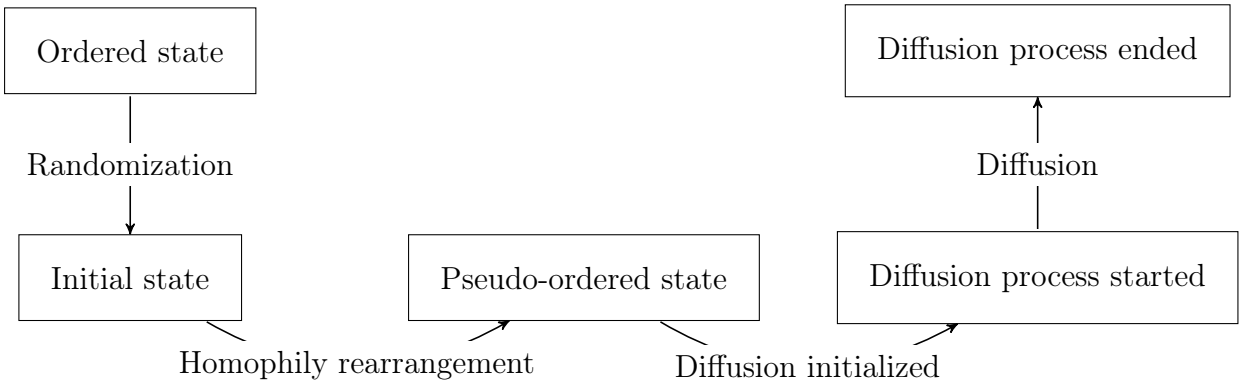


Figure 1.1: The schematics of the modeling framework used in my thesis

A high-level overview of how the modeling elements which are introduced in my thesis are connected to each other is depicted in Figure 1.1. Initially the network has a single generic vertex feature or a number of generic vertex features that are assigned to the nodes based on a rule – the network is either representation of a real world complex system or artificial and resulted from a network generation algorithm. The generic vertex properties can be randomized, by this step the topological vertex characteristics are going to be unchanged. The application of the homophily rearrangement algorithms, results in a network which has the same topology, the distributions of the features are the same, but the respective homophily levels reach certain targets. On the homophily rearranged network susceptible-infected diffusion process starts which converges when all of the ver-

tices reach an infected state. This series of steps – feature randomization, homophily rearrangement and diffusion is repeatable in order to characterize the properties of the spreading phenomena. It should be noted that numerical results that can be obtained from similarity based diffusion on the initial network are useful null-model results which are possible benchmarks of the diffusion results obtained on the rearranged network.

Results presented in the thesis and answers given to the research questions are summarized in the following. The thesis proposes three algorithms to solve the univariate and multivariate homophily rearrangement tasks. These algorithms are capable of rearranging categorical, ordinal, binary, count and continuous features alike. In addition, the simulation results obtained from applying the algorithms revealed certain intriguing empirical regularities. First, simulation results demonstrated that homophily rearrangement solutions are unstable and the resulting generic features are uncorrelated with the topological vertex properties. Second, sensitivity analysis of results demonstrated that the number of iterative steps needed for solving the homophily rearrangement problems is increasing in the level of target homophily. Third, the sensitivity analysis had also proved that as the size of the system increases the number of steps required in order to achieve a target homophily level also increases. Multivariate simulations of homophily rearrangement highlighted that the correlation of generic vertex features helps the homophily rearrangement process. With regard to similarity based diffusion: the similarity based spreading model demonstrated that homophily propagates the spreading of information and consequently heterophily obstructs the diffusion. The simulations about diffusion with discrimination shown that discrimination slows down the spreading, especially when the initial seeder is from the discriminating group.

Before moving to the next chapter I must highlight a few essential points about the terminologies used throughout the paper. In my whole thesis I use certain terms in an exchangeable way to describe the same concept. The expressions node and vertex are used commutable in the whole paper. The term agents is exchangeable with nodes and vertices in the parts about similarity based diffusion. The idioms link and edge are interchangeable with each other in every single chapter. In addition, the terms features and traits are used as equivalents.

The remainder of my thesis is structured as follows. The related literature on previous research is summarized in Chapter 2. Functions that measure homophily on networks

are characterized and defined for continuous and multinomial features in Chapter 3. A novel contribution is made in Chapter 4 with the introduction of three different types of homophily rearrangement algorithms. Multivariate extensions of the homophily rearrangement algorithms are brought on in Chapter 5. The similarity based diffusion model used for simulations on networks is introduced in Chapter 6. Simulation results of the homophily rearrangement algorithms are presented in the sections of Chapter 7. The findings based on the diffusion processes are dealt with in Chapter 8. The thesis concludes by Chapter 9 with a summary and possible policy applications.

The list of notations is enclosed in Appendix A. Pseudo code of the auxiliary algorithms is in Appendix B. The tables are attached in Appendix C while the additional Figures are in Appendix D. The R implementations of algorithms, models and experiments can be found in Appendix E. The R codes were also uploaded to `http://github.com/benedekrozemberczki` to help reproducibility.

Chapter 2: Literature review

Homophily as a social phenomenon has an extensive theoretical and empirical literature. In order to position the homophily rearrangement algorithms and the similarity based diffusion model introduced by my thesis I provide an overview of the relevant homophily related literature. The chapter is divided into three main sections. First, I discuss the general context of homophily and related empirical results in Section 2.1. Various measures of homophily are treated in Section 2.2. The literature on homophilous network generation algorithms, homophily rearrangement algorithms and similarity based diffusion is covered by Section 2.3.

2.1 The context of homophily

The term *homophily* is applied by Lazarsfeld & Merton (1954) to describe the phenomenon that people are more likely to have connections with other people who are similar to themselves. McPherson et al. (2001) uses homophily in a slightly different denotation – their homophily definition has a time dimension, namely that people who are similar to each other interact more frequently. Homophily is even used by Jackson (2010) only to refer to the regularity that in social networks the linked actors tend to be similar to each other. In my thesis the term *homophily* will be used to describe the regularity when vertices in a particular network are more likely to have links with other vertices that are similar to them in generic vertex properties. Another important and closely related phenomenon is heterophily. Heterophily is present in a network when vertices are more likely to have links with other vertices that are different from them regarding generic vertex features. These general definitions of homophily and heterophily allow for a universal understanding of the phenomenon in a wide variety of complex networks – based on these more general definitions one can investigate homophily in more abstract networks such as the network of protein-protein interactions or the network of blogs that spread similar political ideas. It should be emphasized that homophily and heterophily

are macro-level properties of a network that originate from the micro-level similarity of the vertices (Schelling, 1969; Jackson et al., 2016). In addition, my approach assumes that the generic feature similarity among vertices (homophily) is quantifiable and the quantification is distance based (Frey & Dueck, 2007).

Interestingly, in the work of Noldus & Mieghem (2015) the term non-topological feature assortativity is used to describe homophily regarding vertex features. Importantly, the empirical phenomena observed in social networks that Noldus & Mieghem (2015) describe are the same as what the majority of the literature coins as homophily. The term *generic vertex feature assortativity* is used interchangeably with homophily by Quayle et al. (2006), who point out that the term homophily is more widely used in social sciences. In my thesis I use the term homophily to describe the macro-level aggregated similarity of generic vertex properties on every network (not just on social networks).

2.1.1 Homophily on social and economic networks

In order to prove that homophily should be used in a more general sense I collected results of empirical research regarding the presence of homophily within and outside the social sciences. This section deals with socio-economic networks that show homophily. First, I give an overview on the literature connected to homophily in friendship networks. Second, I summarize our current knowledge on corporate governance networks. Third, I epitomize empirical findings about the network of labor market referrals. Finally, the relevant research on the homophilous nature of sexual relationship networks and assortative mating is traversed. It is evident that the covered literature is not exhaustive, there are other socio-economic networks that show homophily or heterophily, but in this section only the above listed networks are considered based on their policy relevance.

Friendship networks

A generic example of social networks is a network of friendships where people are represented by the nodes and the links are the friendships between pairs of people. The fact that friendship networks (and most of the social networks in multiple dimensions) are homophilous is an ubiquitous assertion (Lazarsfeld & Merton, 1954). However, the measured level of homophily originates in two specific social processes, namely *self-selection* and *peer-influence* (McPherson et al., 2001). Self-selection in the social networks setup simply means that people form social relationships with others who share with them certain fairly immutable features such as race or gender. Importantly the linked persons can

influence each others mutable features after bonding – these dimensions probably include political views, preferences and other properties that are generic vertex features. Importantly, this means that the observed homophily regarding features that are mutable is a result of peer-influence among people and self-selection into having a social connection. The fact that two factors contribute to the measured homophily is indicative regarding that cross-section measurement of homophily on social-networks is likely to be biased in case of mutable features.

As the works of Epstein (1986) and Moody (2001) highlight it networks of friendships show strong homophily regarding race, gender and age. While the research of Epstein (1986) and Moody (2001) was only restricted to the investigation of students in the United States it is easy to see that their results are likely to have high external validity. Namely that friendship networks are homophilous with respect to race, gender and age. The before mentioned findings about the network level preferences towards racial, gender and age based similarities of friendships are supported by a wide range of literature (Kao & Joyner, 2004; Shrum et al., 1988; Noel & Nyhan, 2011; Mayer & Puller, 2008).

However, universal homophily with respect to these features does not characterize subnetworks of the system which might not share the properties of the whole-system. It has to be noted that networks of friendships are compositions of friendship clusters, highly connected cliques of closely related friends. Moreover, the same person might belong to multiple friendship clusters and these clusters probably show heterogeneous levels of homophily. The results of Gonzalez et al. (2007) show that in friendship clusters the level of observed homophily might depend on the distribution of features. For example, whites in minority might act differently than blacks in minority. This specific mechanism is also supported by the findings of Wimmer & Lewis (2010).

Corporate governance networks

In corporate governance networks, boards of companies are represented by the nodes and they are linked by the board members who are present in both boards. Essentially these networks are projected versions of bipartite networks where one type of nodes is represented by companies, and the other type is represented by the board members. As Kogut et al. (2012) point out these networks show strong homophily in multiple dimensions. These dimensions include the average experience and tenure of the boards members, the industry of the firm and the gender composition of the boards that are

interlinked. The fact that companies that are connected by board members are more likely to be in the same industry is not surprising if one considers that board members are likely to have industry specific domain-knowledge and experience. For corporate governance networks in Scandinavia (namely Denmark, Norway and Sweden) Edling et al. (2012) identified similar patterns of homophily. The diverse boards (regarding age and gender) are more interlinked with other diverse boards while homogeneous boards, which consist mostly elderly males, are usually linked with other homogeneous boards. This simply means that Scandinavian corporate governance networks show homophily regarding the board homogeneity.

Labor market referral networks

In labor market referral networks the referees and refereed persons represent nodes, and the referral itself is the link between them. As Petersen et al. (2000) shown the labor market referrals are biased towards sharing a common gender and race on the labor market of the United States. However, their results did not support evidence that the hiring process itself would be homophilous towards the above mentioned variables. In their similarly designed research, Fernandez & Fernandez-Mateo (2006) investigated the network of labor market referrals of a medium sized company in the United States and their results support the findings of Petersen et al. (2000). On the supply side they found that the referee and the refereed person have the same gender and race with a disproportionate probability – which is a sign of supply side homophily. Interestingly, about the demand side they found that during the screening and selection there is no evidence of homophily or heterophily towards race. However, Fernandez & Fernandez-Mateo (2006) observed that there is weak evidence of a gender based homophily during the screening process – male interviewers favor disproportionately males.[1]

Networks of sexual relationships

The network of sexual relationships is driven by the assortative mating phenomenon – namely the principle that romantic relationships are between participants who are on average similar to each other. Formally, this definition of assortative mating describes that the network emerging from the pairwise relationships is homophilous. The assortative

[1] Although it is beyond the scope of my thesis, but it must be noted that these empirical regularities found by Petersen et al. (2000) and Fernandez & Fernandez-Mateo (2006) together with the assumption of homophilous network formation and the model of Calvo-Armengol & Jackson (2004) have troublesome implications about the effects of homophily on the employment chances of minorities and females.

mating extends to the education level of couples (Mare, 1991), their ethnic background (Vandenberg, 1972), shared hobbies (Kalmijn & Flap, 2001) and even to their aligned political views (Watson et al., 2004). Because of the above mentioned facts, namely that the structure of sexual networks is affected by homophily, investigating the nature of homophily and diffusion in sexual networks helps understanding why certain communities are more affected by sexually transmitted diseases. The work done by Laumann & Youm (1999) revealed that the network of sexual relationships in the United Sates of America shows homophily regarding education, income and race. In their work Laumann & Youm (1999) demonstrated that the increased prevalence of HIV in the African-American community has roots in the interplay of the sexual network's topology and the distribution of features. First, the Africa-American community shows a higher level of homophily than any other community in this regard. Second, the peripheral African-Americans are more likely to have sexual relationships with those who are in the core of African-American sexual relationship network. This is not true for other races – peripheral whites have sexual relationships with peripheral whites and peripheral Hispanics and Asians act similarly.

Importantly the results of Bearman et al. (2004) on the sexual relationship network of teenagers in the United States point out an empirical phenomenon that later introduced homophily rearrangement algorithms can reproduce. Besides the homophilous nature of the network regarding features such as test scores and income, Bearman et al. (2004) also demonstrate that students who have rare feature values are on the periphery in the sexual relationships network. First of all as I mentioned this is reproducible and also it implies that vertices with extreme properties are less prone to contagion.

According to Kenyon & Colebunders (2013) the network of sexual relationships in Sub-Saharan Africa exhibits heterogeneous levels of homophily regarding socio-economic features such as ethnic group, race, age and education level. They found that sexual relationships of larger ethnic groups show strong homophily, while small size groups show heterophily – which is meaningful if one considers the risk of inbreeding in a population. It should be noted that in Kenyon & Colebunders (2013) certain generic features such as education level, income and race are positively correlated which shows that homophily might be endogenous in this specific network.

2.1.2 Homophily on non socio-economic networks

As it was previously pointed out not only socio-economic networks show homophily. A number of non socio-economic networks show homophily which fosters the idea that homophily is not just a social phenomenon, therefore in this subsection I give an overview about such networks. First, I will give computer networks as an example. Second, the network of hyperlinked blogs is used as an illustrative homophilous non socio-economic network. Finally, the homophilous nature of protein-protein interactions is highlighted.

Computer networks

The idiom computer networks is unequivocal, it simply means that computers are represented by nodes in the network and if two of them are connected they will share a link – the nodes and links together form a network. The main ideas outlined in Balthrop et al. (2004) underpin that computer networks show homophily regarding multiple generic vertex features such as quality of hardware, specific software versions, firewalls and patches applied to the softwares. An unpretending implication coming from Balthrop et al. (2004) that the before mentioned homophilous network properties is that shared features of clusters that make computers vulnerable to targeted attacks make the whole cluster vulnerable to these attacks. This simply means that a shared feature of vertices might propagate diffusion and spreading. The similarity based diffusion model that I introduce in Chapter 6 is capable of reproducing this regularity, the fact that homophily might help the diffusion of a contagion.

Networks of hyperlinked blogs

As Park & Thelwall (2003) points out, blogs on the web are likely to be connected to other blogs that deal with similar content. In their small-scale examination of political blogs in South-Korea, Park et al. (2001) demonstrated that the hyperlink network of political blogs and news pages is homophilous regarding political views. It is unequivocal that this is not just a phenomena which has internal validity only. The results of Bisgin et al. (2010) support the idea that the findings of Park et al. (2001) can be generalized. Based on data from *BlogCatalog* which is an American blog service provider, Bisgin et al. (2010) prove that political blogs have hyperlinks to other blogs that spread similar political ideas. Furthermore, Bisgin et al. (2010) also revealed that the hyperlinked blogs on *BlogCatalog* exhibit homophily with respect to their general category – for example, entertainment blogs are mostly connected to other entertainment blogs.

I have emphasized in the case of social networks that the measured level of cross-section homophily has two root causes – the initial selection into attaching a link and the apparent influence between connected nodes. Roth & Cointet (2010) describe a similar phenomenon regarding the hyperlinked blogs and the content on these blogs. Their study uses data on the network of political blogs in the United States and the content on these political blogs. First, blogs establish hyperlinks to other blogs that have a similar content. Simply the attachment rule describing the linking is preferential towards the content, which results in homophily. Later one observes an increasing similarity between the connected blogs semantical content, due to influence between the hyperlinked blogs.

Biological networks –protein-protein interactions

Protein-protein interaction networks are specific examples of so called *intractomes*, in these networks vertices represent the molecules (here specifically proteins) and edges are representing the interactions among molecules. As Rahmani et al. (2011) points out, the understanding of homophily on protein-protein interaction networks helps effective targeted medicine design and identification of proteins that are associated with diseases. A major concern about the before mentioned usefulness of protein-protein interaction networks is that for certain proteins their functionality and association with a disease (generic vertex features) are unknown and probabilistic predictions about these features have to be made. It is worth noting that unlike the majority of networks brought forward in my thesis the protein-protein interaction networks show both homophily or heterophily with respect to the function of the proteins (Rahmani et al., 2011). In addition, Rahmani et al. (2011) argue that due to the co-existence of homophilous and heterophilous protein-protein networks implies that on different networks different types of predictive methods will be effective (e.g., support vector machines, random forests or neural networks). Practically, this means that the type of predictive analytics used to decide the functionality of proteins has to be hand-picked for every specific protein-protein interaction network.

The extensive work of Navlakha & Kingsford (2010) highlights the homophilous nature of protein-protein interaction networks with respect to being associated with a certain disease. They show that proteins that are associated with a certain type of diseases are more interconnected with each other. Additionally, they revealed that on protein-protein interaction networks the increase of homophily in absolute terms is also increasing the accuracy of predictive methods. This also means that identifying proteins that are

associated with the disease is easier when homophily (or heterophily) is stronger in the network.

2.2 Measuring homophily and segregation

A central question in my thesis is how one could control the level of homophily in a complex network. In order to control the level of homophily, it has to be measured with a function that describes homophily and target homophily values have to be predefined according to the chosen homophily measurement function. This section gives an overview on the most frequently used homophily measures in the literature.

If one looks at the homophily measurement related literature it is clear that there are two main typification of homophily measurement functions. First, a categorization can be made based on the types of generic vertex features that one can process with the measure. In this regard there are homophily measurement functions that measure homophily regarding categorical, ordinal and binary features and their output is a vector of group specific homophily values – later from Chapter 3 I reference these homophily measurement functions as *categorical homophily measurement functions*. Other homophily measurement function are able to characterize homophily regarding continuous generic vertex features – these are upheld as *universal homophily measurement functions*. Second, the categorization can also be made based on whether the function gives an overall system level measure of homophily or rather individual vertex specific measure of homophily (or segregation). The algorithms introduced by the thesis deal with homophily measurement functions that consider both binary, count, categorical, ordinal and continuous features which provide systematic measure of similarity in the network. In the forthcoming paragraphs I characterize the most important measures of homophily in the literature – a summary table with the measurement functions is enclosed in Appendix C as Table C.1.

A frequently used measure of categorical homophily is the *homophily index* introduced by Coleman (1958). This measure gives a measure of homophily or heterophily for each of the vertices that share a common feature value regarding a categorical, ordinal or binary feature. Importantly, this measure is constrained to the $[-1, 1]$ interval and calculated from the within group average degree (vertices that share a common generic vertex feature value) and outside the group average degree (vertices that do not share a common generic vertex feature value). In their work Currarini et al. (2009) go a step further and

normalize Coleman's homophily index with the relative size of the group. This modification of the original homophily index takes into account that the relative group size is a serious constraint on the potential level of homophily if the average degree in the different groups (vertices that share a common generic vertex feature value) is independent from the group membership.

Another categorical homophily measurement function which is comparable to Coleman's homophily index is the *external-internal link index*, which was introduced by Krackhardt & Stern (1988). The external-internal link index can measure homophily in respect to categorical, ordinal and binary generic vertex features. It gives for each distinct value of the feature a groups specific homophily level in the $[-1, 1]$ interval. This normalization of the measurement (the constrained interval and zero expected value) means that homophily levels regarding different features are comparable on the same network. If one takes a closer look, we can see that the external-internal link index is directly related to Coleman's homophily index – the difference lies in that the external-internal link does not divide the number of links with the number of vertices.

The *segregation matrix index* described by Freshtman & Gneezy (2001) is an additional categorical homophily measure which is constrained to the $[-1, 1]$ interval. This metric measures homophily based on the within and out of group density in the network if the feature is binary, ordinal or categorical. For each group it assigns a value based on the difference of the within and out of network density, normalized by the sum of the before mentioned densities. By its construction it has similar meaning as the external-internal link and the homophily index. If it has a negative value the groups which share a common feature value (and which is investigated) show heterophily if it has a positive value they show homophily.

It is straightforward that averaging out homophily measures that describe group (or individual) specific homophily levels would result in measures that describe system wide homophily. However, in my whole thesis I do not consider such approaches, because taking the arithmetic mean would obscure the group specific behaviors. Moreover, certain categorical and individual homophily measures such as the *spectral homophily index* (Echenique & Fryer, 2007), the homophily test (Easley & Kleinberg, 2010) or the *Gupta-Anderson-May index* (Gupta et al., 1989) do not take values from a constrained interval. So averaging them without rescaling would not be meaningful. To sum it up, my thesis

does not consider homophily measures that give individual level measure of segregation or homophily and categorical homophily measures are not going to be averaged.

On universal homophily measurement functions, a frequently used universal homophily measure is the *segregation index* of Freeman (1972, 1978). However, the original formulation of this measure only considers a single binary variable. Because of this in the algorithm design process I am not going to apply it. This measure gives a single network wide measure of homophily in the $[0, 1]$ interval. It takes zero when the networks shows baseline homophily (random assignment of edges) and one when the network shows perfect homophily. A simplistic measure of system-wide universal homophily can be Pearson's linear correlation coefficient if it is calculated based on all of the node-pairs. As Noldus & Mieghem (2015) point out if one considers that every edge has two endpoints (the vertex pair) and these endpoints have feature values, listing all the features of edge endpoints results in two variables. These two variables can be correlated with each other. While Noldus & Mieghem (2015) only considers linear correlations, other correlation measures can be computed from such variables with ease. These correlation based measures are all universal homophily measurement functions as they do not describe group specific homophily levels. Because these are computationally cheap measures of homophily they are possibly useful for algorithm design.

The *spatial autocorrelation measures* introduced by Moran (1950) and Geary (1954) are actually measures of homophily regarding continuous, count and binary variables. These spatial autocorrelation measures take into account feature values in the first order neighborhood's of vertices in a network which has an arbitrary topology and give back measures of system wide homophily. In addition, they can be computed based on the adjacency matrix of the network and the generic vertex feature vector which makes them useful in later algorithm design efforts of my thesis.

2.3 Homophilous network generation, homophily rearrangement and similarity based diffusion

In this section first I discuss the ideas of *homophilous network generation* and *homophily rearrangement*. Moreover, I distinguish between them, the two concepts are similar, but only the second one is a core concept investigated in my thesis. After this distinction is made, I summarize the main ideas and results about homophilous network generation,

homophily rearrangement and the connection between homophily and diffusion.

Homophilous network generation means that one generates a network that shows homophily either by a probabilistic network generation process or by rearranging links of a network among the vertices. In contrast, homophily rearrangement means that generic vertex feature values are exchanged in order to achieve a given level of homophily. Importantly, in case of the first concept the topology is not fixed, while in the second case the topology cannot be changed.

One of the major novel modeling approaches to generate homophilous networks is when the homophily is a result of a micro-level similarity based preferential attachment rule. In these models, the network starts from a state when vertices have no links among them. The probability of realizing a linkage depends positively on the similarity of vertices (regarding generic vertex features). With simplistic assumptions, these models are able to show that homophily can be a result of preferences towards similarity (Jackson et al., 2016). These models are different from the probabilistic network generation models that Barabási & Albert (1999) and Bianconi & Barabasi (2001) contrived, because the preferential attachment is purely generic vertex feature similarity based. A common element of these models is that they are able to reproduce a number of empirical regularities, namely:

1. Strong preferences towards similarity in a variable result in a strong homophily regarding the variable in the resulting network.

2. The stronger the vertices preference towards similarity, the stronger is clustering in the resulting network.

3. The degree assortativity is increasing as the preferences towards similarity increase.

The preferential attachment model that Quayle et al. (2006) propose uses generic vertex feature similarity to generate scale-free networks that show homophily. With their simplistic approach they were able to prove that homophily increases as the preferential attachment is more similarity based (similarities are upweighted), and that generic feature correlations increase within communities as the preference towards similarity increases. However, Quayle et al. (2006) made no attempt to supervise the macro-level similarity in the model, there is no exact control of the overall level of homophily in their model.

Another similarity based preferential attachment model is the one that van Eck & Jager (2010) introduced. Their similarity based preferential attachment algorithm generates networks that show homophily regarding multiple variables. They use survey data obtained from social media to describe the individual preferences of agents in their model. In the initialization of their model, the agents have no social ties, and later social ties form based on individual preferences. In this regard (micro-level incentives) their model is similar to the one that Quayle et al. (2006) implemented and the greedy homophily rearrangement algorithm that I propose later in Chapter 4. They also investigate certain macro-level characteristics of the system such as macro-level satisfaction (utility) of the agents. However, their model has no macro-level control of homophily (there is no target homophily value). Essentially the above mentioned facts mean that the network generator processes introduced by Quayle et al. (2006) and van Eck & Jager (2010) are fundamentally different from the models that Yavas & Yusel (2014) and my thesis introduces.

The other major approach to generate networks that show homophily is to generate an initial network and later rearrange links in a way that a target homophily criteria is satisfied. A homophilous network generator model is the one that Holzhauer et al. (2013) inducted in order to generate homophilous agent based modeling setups. However, like in many other approaches, Holzhauer et al. (2013) does not generate networks that show homophily regarding multiple variables. The agent based model and the framework that Yavas & Yusel (2014) proposes is suitable for investigating the relationship between homophily and diffusion. In their analysis Yavas & Yusel (2014) used a two stage model: first they generated networks which had shown a given level of homophily. Later they started diffusion processes on the networks which were influenced by similarity of the connected vertices. My approach is also two-stage, but unlike the generator process that Yavas & Yusel (2014) introduced, my homophily generator process assumes that the network structure is fixed and only the generic vertex features are exchangeable. The diffusion stage is also different in my approach, because I modify the susceptible infected model, while the authors used a modified variant of the threshold model. The reason that I chose the susceptible-infected model is that it can be modified with ease to characterize similarity based diffusion.

The agent-based segregation model of Schelling (1969) might be classified as a homophily rearrangement algorithm, because the topology of the network where agents are allocated is totally fixed. However, there are three considerable weaknesses of this model.

First, the agents are located on a square lattice which is a very simplistic topology – the emerging homophily . Second, there is no exact macro-level control over the homophily appearing in the system – the homophily is a byproduct of the micro-level incentives of agent's. Third, in the basic model a number of vertices in the network are *vacant* – so agents do not change their location with each other, but they simply move to the vacant vertices that are in line with their individual preferences. The first potential concern was addressed by Fagiolo et al. (2007) who investigated the model proposed by Schelling (1969) on networks that show a non-uniform degree distribution. Their results with this modified model supported that micro-incentives might lead to emerging homophily on canonical complex networks. Moreover, augmenting the Fagiolo et al. (2007) model with an early stopping condition (regarding system-level homophily) would make their model a homophily rearrangement algorithm. Nevertheless, the problem of vacant vertices would remain and the greatest novelty of the homophily rearrangement algorithms proposed in my thesis lies with respect to the fact that there are no vacant vertices in the system.

The fact that individual preferences towards similarity and homophily might affect diffusion is widely researched. At the same time, the broad research done concerning the relationship of homophily and diffusion is rather inconclusive. There are mainly two conceptions about how homophily effects diffusion: one is that it slows down the spreading compared to a none homophilous state the other is that it speeds up the diffusion process. Intriguingly, both of these ideas have reasonable theoretical models and literature of empirical evidence that supports them. The simulations of similarity based diffusion in my thesis will show that the homophily affects diffusion heterogeneously.

First, theoretical model of Golub & Jackson (2012) had shown that under certain conditions homophily might slow down the diffusion of information on a network (learning is slackened). This ideas is supported by the empirical findings of Kamath & Cowan (2015), who shows that homophily can be an obstacle to the diffusion of technological innovations when innovations appear in dense clustered communities and extra-group relationships are rare in the observed network.

Second, the agent-based model introduced by Yavas & Yusel (2014) promotes that homophily can speed up diffusion on a wide-range of networks. The experiments of Centola (2011); Centola & van de Rijt (2015), who investigated the adaptation of healthy life-style related behaviors, backed up the theoretics of Yavas & Yusel (2014) that homophily prop-

agates diffusion. This positive relationship between homophily and diffusion is supported by the non-experimental findings of de Matos et al. (2014) who shown that homophily positively influences the adaptation of information technologies. Importantly, Centola (2011) uses a small-scale experimental setup where the topology is fixed while the homophily can be controlled arbitrarily. With this unique experimental setup peer-effects and the effect of homophily on health-related decisions becomes quantifiable. This fixed topology and controlled homophily approach will characterize my later efforts – as my goal is to rearrange features in a homophilous way on a network that has a given topology while the diffusion is affected by homophily.

The spreading model that I develop is based on the *susceptible infected model* and different from the *independent cascade model*, because the same node might try to pass the infection (information) multiple times to the neighboring nodes. In case of the independent cascade model each of the nodes has a single probabilistic attempt to propagate the infection to other nodes. Because of this, the heuristic quasi-optimal initial seeding method of Kempe et al. (2003) which helps to maximize the potential spreading is not applicable for the similarity based diffusion model.

Chapter 3: Homophily measurement

Let us imagine that we have a network with a single generic vertex feature denoted by $\mathrm{G}(V, E, \mathbf{x})$. In this network V is the set of vertices, E denotes the set of edges among the vertices and $\mathbf{x}$ represents a single generic vertex feature. Another representation of $\mathrm{G}(V, E, \mathbf{x})$ is the network $\mathrm{G}(\mathbf{W}, \mathbf{x})$, where $\mathbf{W}$ is the adjacency matrix and $\mathbf{x}$ is the single generic feature vector. This simple network might represent students in a high school (vertices), their friendships (edges) and properties of students such as age, gender or race. The phenomenon of interest in this chapter is the homophily regarding the single generic vertex feature and the proper measurement of homophily.

This chapter introduces the fundamental definitions of homophily measurement needed for the implementation of homophily rearrangement algorithms. To illustrate the concepts introduced in the chapter I included a number of detailed examples. Section 3.1 introduces the concept of universal homophily measurement functions, which are used extensively in the algorithm designs later in my thesis. Categorical homophily measurement functions are discussed in Section 3.2 and ensemble homophily measurement and related concepts are developed in Section 3.3.

3.1 Universal homophily measurement

Definition 3.1. *The universal homophily measurement function $\mathcal{H}_U(\mathbf{x}, \mathbf{W}) \to c$ takes the feature vector $\mathbf{x}$ and the weight matrix $\mathbf{W}$ of the network and gives a measure of homophily (here denoted by c) on the network. If the number of elements in the feature vector is denoted by N, the universal homophily measurement function is essentially a $\mathcal{H}_U(\mathbf{x}, \mathbf{W}) \to \mathbb{R}$ mapping, where $\mathbf{x} \in \mathbb{R}^N$ and $\mathbf{W} \in \mathbb{R}^{N \times N}$. A universal homophily measurement function requires a feature vector $\mathbf{x}$ that is a binary, count or continuous variable.*

The universal homophily measurement function described in definition 3.1 satisfies the following requirements:

1. Takes a positive value, when the network shows homophily regarding variable $\mathbf{x}$.
2. It takes a negative value, when the network in $\mathbf{x}$ shows heterophily.
3. It is zero when the network shows neither heterophily or homophily in variable $\mathbf{x}$.

These formal requirements, for example, imply that the measure described by Moran (1950) (a metric used for describing spatial autocorrelation) can be considered to be a universal homophily measure. It is zero when homophily or heterophily regarding the feature is not present in the network, takes negative value in presence of heterophily and positive when the network shows homophily. The formula used for calculating Moran's I is described by Equation (3.1). The notations used in the equation are in line with Definition (3.1), the only new element is $\overline{\mathbf{x}}$, which describes the mean of the feature vector. It should be noted that the number of operations needed for calculating the numerator of the second fraction in Equation (3.1) scales quadratically with the number of vertices – this means that for larger networks the computation of Moran's I is demanding.

$$\text{Moran's I} = \frac{N}{\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{W}_{ij}} \cdot \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{W}_{ij}\cdot(\mathbf{x}_i-\overline{\mathbf{x}})(\mathbf{x}_j-\overline{\mathbf{x}})}{\sum_{i=1}^{N}(\mathbf{x}_i-\overline{\mathbf{x}})^2} \tag{3.1}$$

Intriguingly, the network level segregation index introduced by Freeman (1978) cannot be considered to be a universal homophily measurement function. It should be noted that another spatial autocorrelation measure called Geary's C does not satisfy any of these requirements, so it cannot be considered to be an homophily measure function (Geary, 1954). However, some of the linear transformations of this specific function satisfy these prescribed requirements. In Equation (3.2) I define a measure that is based on Geary's C to express a function that is a universal homophily measurement function.

$$\text{Reciprocal Geary's C} = 1 - \frac{(N-1)\cdot\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{W}_{ij}\cdot(\mathbf{x}_i-\mathbf{x}_j)^2}{2\cdot\left(\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{W}_{ij}\right)\cdot\sum_{i=1}^{N}(\mathbf{x}_i-\overline{\mathbf{x}})^2} \tag{3.2}$$

Definition 3.2. *A universal homophily measurement function $\mathcal{H}_U(\mathbf{x},\mathbf{W})$ is normalized when it satisfies that $\mathcal{H}_U(\mathbf{x},\mathbf{W}) \to \{c \in \mathbb{R} \mid -1 \leq c \leq 1\}$.*

Essentially definition 3.2 implies that a universal homophily measurement function is normalized when its value is constrained to the $[-1, 1]$ interval. A normalized universal homophily measurement function is Moran's I or the reciprocal Geary's C defined by Equations (3.1) and (3.2), respectively. This property of a universal homophily measurement function ensures that the level of measured homophily will be scale invariable. For example, in a network of firms connected by supplier relationships, the measured homophily in market capitalization will be unchanged by the rescaling of market capitalization from dollars to million dollars. This setting of lower and upper limits also helps in the understanding of perfect heterophily and homophily.

Remark 3.3. *The universal homophily measurement functions defined by Equations* (3.1) *and* (3.2) *require that the feature vector* $\mathbf{x}$ *has an arithmetic mean. Definition 3.1 allows for binary features, in case of binary features the arithmetic means represents the ratio of vertices with a feature value equal to one. If* $\mathbf{x}$ *is an ordinal variable, it must have a numeric variable representation which has an arithmetic mean.*

An ordinal feature with meaningful arithmetic mean can be the highest school qualification of a person, the highest school qualification can be equivalent to the overall number of years associated with obtaining a degree. Therefore the strings can be mapped into numeric values – for details see Equation (3.3).

$$\text{Schooling} = \begin{cases} 0, \text{ if did not attend school.} \\ 8, \text{ if highest degree is elementary school.} \\ 12, \text{ if highest degree is high school.} \\ 15, \text{ if highest degree is a Bachelor's degree.} \\ 17, \text{ if highest degree is a Master's degree.} \\ 22, \text{ if highest degree is a PhD.} \end{cases} \tag{3.3}$$

However, not every ordinal variable has similar mappings like this. For example, if unfinished educational degrees are included, the mapping from ordinal variable to a purely numerical one is ambiguous. Another example of ambiguous mapping can be the case of a language knowledge ordinal feature (with feature values basic, intermediate, advanced and native) into numeric values. There is no reliable way to map with a fair reliability such variable into a numeric one.

Definition 3.4. *A network shows perfect universal homophily in feature* $\mathbf{x}$ *when* $\mathcal{H}_U(\mathbf{x}, \mathbf{W})$ *is a normalized universal homophily measurement function and* $\mathcal{H}_U(\mathbf{x}, \mathbf{W}) = 1$.

Definition 3.5. *A network shows perfect global heterophily in feature* $\mathbf{x}$ *when* $\mathcal{H}_U(\mathbf{x}, \mathbf{W})$ *is a normalized universal homophily measurement function and* $\mathcal{H}_U(\mathbf{x}, \mathbf{W}) = -1$.

The concept of perfect heterophily and homophily can be understood through visualizations and a simple example. Let us imagine that we have a network of firms that are linked by supplier relationships – this is depicted in Figure 3.1. The firms (vertices in the network) are located on a square lattice without periodic boundary conditions. The ones that are located on the corners of the lattice have 2 connections, the ones on the sides but not on the corners have 3 neighbors and the remainder has 4 neighbors. The firms have one single feature, their capitalization. There are two types of firms in this simple system – Type I. has a capitalization of \$100, while Type II. has a somewhat higher capitalization, namely \$200. In Figure 3.1 firms from Type I. are shown as black nodes, while Type II. firms are shown as white ones. The size of the lattice is 4×4 which means that this simple system has 16 vertices and 24 edges.

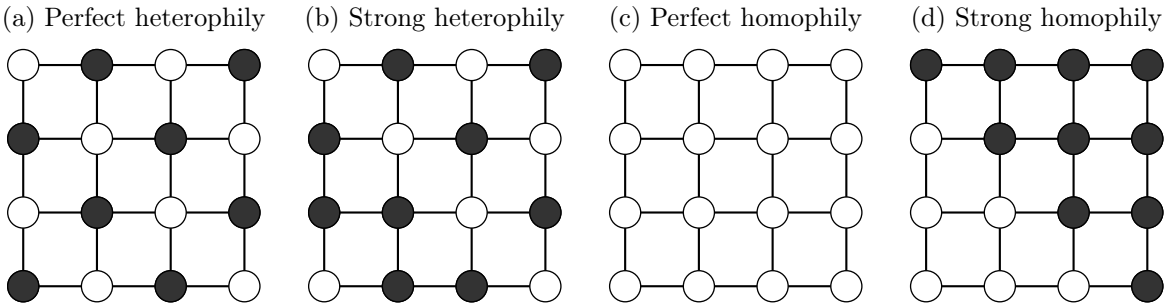


Figure 3.1: Different levels of universal homophily on a 4×4 square lattice without periodic boundary conditions

The subfigures of Figure 3.1 show four different levels of homophily, for all four cases I measure the level of homophily with universal homophily measurement functions. The perfect heterophily is shown by Subfigure 3.1a. For this setup the value of Moran's I, the linear transformed Geary's C and the end-node correlation are the same, namely, all of them are -1. In Subfigure 3.1c perfect universal homophily is shown, the respective measures all have a value of 1. This artificial example demonstrates that perfect homophily can exist only in two scenarios:

1. There is only one value that the generic feature of the vertices can take.
2. Vertices who share a common vertex value are segregated and do not have connections with vertices who have a different feature value.

A state with strong homophily is depicted in Figure 3.1d, where the end node correlation is $0.46\dot{6}$, Moran's I is $0.49\dot{4}$ and the linear transformed Geary's C is $0.48\dot{3}$. While these values are close to each other, they are incomparable.

3.2 Categorical homophily measurement

The measurement approach discussed in Section 3.1 is only viable when the generic vertex feature $\mathbf{x}$ is not categorical. However, in a number of networks the generic vertex features are categorical – race, religion or social status in social networks are such generic vertex features. Consequently, to characterize homophily related to categorical variables a slightly different set of definitions has to be introduced.

Definition 3.6. *The categorical homophily measurement function $\mathcal{H}_C(\mathbf{x}, \mathbf{W}) \to \mathcal{C}$ takes the feature vector $\mathbf{x}$ and the weight matrix $\mathbf{W}$ of the network and gives a measure of homophily (here denoted by $\mathcal{C}$) on the network. If one denotes by N, the number of elements in the feature vector, the universal homophily measurement function is essentially a $\mathcal{H}_C(\mathbf{x}, \mathbf{W}) \to \mathbb{R}^M$ mapping, where $\mathbf{x} \in \mathbb{R}^N$ and $\mathbf{W} \in \mathbb{R}^{N \times N}$. The M denotes the number of unique values that a categorical feature vector $\mathbf{x}$ can take. A categorical homophily measurement function requires a generic vertex feature which is categorical (or ordinal).*

Essentially the homophily is measured for each of the groups and $\mathcal{C}$ is a vector with M components. The categorical homophily measurement function described by Definition 3.6 has the following properties:

1. The group specific homophily measurement value $\mathcal{C}_m$ is positive when vertices which have an $\mathbf{x}$ value equal to m show homophily.

2. The group specific homophily measurement value $\mathcal{C}_m$ is zero when vertices which have an $\mathbf{x}$ value equal to m neither show heterophily or homophily.

3. The group specific homophily measurement value $\mathcal{C}_m$ is negative when vertices which have an $\mathbf{x}$ value equal to m show heterophily.

This proposed definition allows that regarding that vertices which share the same $\mathbf{x}$ value might show heterophily, while other vertices with different $\mathbf{x}$ might show homophily. For example, the study of Kenyon & Colebunders (2013) shown that the network of sexual relationships might show heterophily for certain small size ethnic groups and homophily for other ethnic groups that are relatively large.

Definition 3.7. *A categorical homophily measurement function* $\mathcal{H}_C(\mathbf{x}, \mathbf{W})$ *is normalized when it satisfies that* $\mathcal{H}_C(\mathbf{x}, \mathbf{W}) \rightarrow \{\mathcal{C} \in \mathbb{R}^M \mid -1 \leq \mathcal{C}_m \leq 1, \quad \forall m \in M\}$.

The homophily index of Coleman (1958) and the segregation matrix index of Freshtman & Gneezy (2001) are both normalized categorical homophily measurement functions according Definition 3.7. The way that Definition 3.7 introduces normalized categorical homophily measurement functions is a straightforward extension of Definition 3.2. The level of categorical homophily is constrained to the $[-1, 1]$ interval for each of the groups. Homophily measures such as the homophily, inbreeding homophily and segregation indices are all normalized categorical homophily measurement functions. Importantly, the normalization of the homophily rearrangement function makes homophily levels across groups comparable. One can compare the level of ethnic homophily (specifically for Asians, Blacks, Hispanics and Whites) in a network of friendships or the level of homophily among linked blogs or websites regarding political affiliation.

Definition 3.8. *A network shows perfect categorical group homophily regarding vertices that have feature* $\mathbf{x} = m$ *when* $\mathcal{H}_C(\mathbf{x}_m, \mathbf{W})$ *is a normalized categorical homophily measurement function and* $\mathcal{C}_m = 1$.

Definition 3.9. *A network shows perfect categorical group heterophily regarding vertices that have feature* $\mathbf{x} = m$ *when* $\mathcal{H}_C(\mathbf{x}_m, \mathbf{W})$ *is a normalized categorical heterophily measurement function and* $\mathcal{C}_m = -1$.

Definitions 3.8 and 3.9 take into account that groups that have a certain level of a categorical feature $\mathbf{x}$ might show perfect homophily or heterophily while groups with different levels might show a non-perfect level of heterophily or homophily. This definition also implies that vertices that show perfect categorical group homophily only have relationships within the group, while others who belong to a group that does not have perfect level of homophily have links to vertices that are outside the group. Similarly, it also means that vertices which belong to a given group and show perfect categorical group heterophily have only relationships with nodes that have different $\mathbf{x}$ values. The above mentioned fact has an interesting consequence about universal homophily measurement. Let us imagine that we have a network of friendships, and $\mathbf{x}$ denotes a generic feature which describes the income of the people in this simple network. If one would measure universal homophily regarding income, the network might show homophily. However, if the income variable is binned into a categorical variable, groups that have a small relative size in the population might show perfect categorical group heterophily.

Definition 3.10. *A network shows perfect global categorical homophily regarding feature* $\mathbf{x} = m$ *when* $\mathcal{H}_C(\mathbf{x}_m, \mathbf{W})$ *is a normalized categorical homophily measurement function and* $\mathcal{C}_m = 1, \quad \forall m \in M$.

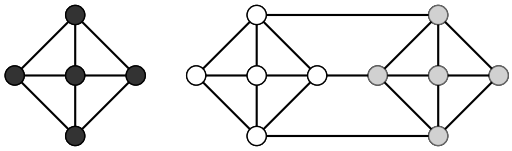
Definition 3.11. *A network shows perfect global categorical heterophily regarding feature* $\mathbf{x}$ *when* $\mathcal{H}_C(\mathbf{x}_m, \mathbf{W})$ *is a normalized categorical heterophily measurement function and* $\mathcal{C}_m = -1, \quad \forall m \in M$.

It is an outright extension of Definitions 3.8 and 3.9 to describe networks that show perfect categorical group homophily or heterophily for all unique values of $\mathbf{x}$. Perfect global categorical heterophily, that is described by Definition 3.10, practically means that vertices which share the same value of categorical feature vector $\mathbf{x}$ have only links to vertices which have the same value of $\mathbf{x}$. Similarly, perfect global categorical heterophily means that it will be true for all vertex, which have $\mathbf{x} = m$ that they only have links to vertices which have $\mathbf{x} \neq m$.

Remark 3.12. *The heuristic explanation of perfect categorical group homophily and perfect global categorical homophily means that for networks which have a single component a state of perfect categorical group homophily cannot be achieved for any group of vertices if* $\mathbf{x}$ *can take more values than one – intrinsically if* $m > 1$. *This also results in the regularity that a state of perfect global categorical homophily does not exist. Practically, a homophily rearrangement algorithm which has a a target of perfect categorical group homophily and runs on a single component network is not going to converge (if one assumes fixed topology).*

The distinction between perfect categorical group homophily/heterophily and perfect global categorical homophily/heterophily can be understood through an example. In the subfigures of Figure 3.2 two simple networks are depicted. Each of the networks has 12 nodes and a single categorical generic feature vector $\mathbf{x}$. This feature can take three distinct values, respectively this is represented as black, gray and white vertex coloring on the network. In Subfigure 3.2a black vertices exhibit perfect categorical homophily, while white and gray ones only show categorical homophily. Thus, there is no perfect global categorical homophily in the system. By contrast, in Subfigure 3.2b the black, white and gray vertices only have within group links, so this network shows perfect global categorical homophily.

(a) Perfect categorical group homophily



(b) Perfect global categorical homophily

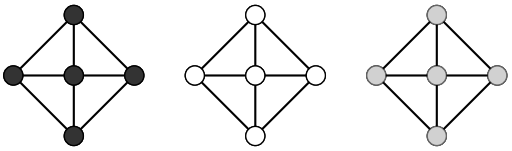


Figure 3.2: Perfect categorical group homophily and perfect global categorical homophily distinction – illustrative example

3.3 Ensemble homophily measurement

In real networks vertices usually have multiple generic features. For example, in a network of friendships the vertices have features such as age, gender, race, income and religion and other variables, among many others. Each of these generic features is a potential driver of homophily, and homophily is measurable regarding all of these variables. Let us imagine that we have a network, and the number of generic vertex features is j, and vertex features are simply denoted as $\mathbf{X}^1, \ldots, \mathbf{X}^p$. Among these generic vertex features one might have multiple types including binary, categorical, ordinal and continuous ones. Together these column vectors can form the generic vertex feature matrix $\mathbf{X}$, which has the individual generic vertex features $\mathbf{X}^1, \ldots, \mathbf{X}^p$ as its columns. For each of the features, homophily is measurable with an arbitrarily chosen homophily measurement function $\mathcal{H}(\mathbf{x}, W)$ – the lower index is missing because the function is either universal or categorical.

Definition 3.13. *The function* $\mathcal{E}(\mathcal{H}_1(\mathbf{X}^1, \mathbf{W}), \ldots, \mathcal{H}_p(\mathbf{X}^p, \mathbf{W}))$ *is an ensemble homophily measurement function if it is a series of* p *arbitrary universal or categorical homophily measurement functions which describe homophily for each features in matrix* $\mathbf{X}$ *with* p *features.*

The ensemble homophily measurement function described by Definition 3.13 can be explained in plain words as follows: we have p homophily measurement functions and they all give a measure of homophily regarding a certain feature, just as Equation system (3.4) shows.

$$\begin{aligned} \mathcal{C}_1 =& \mathcal{H}_1(\mathbf{X}^1, \mathbf{W}) \\ &\vdots \\ \mathcal{C}_p =& \mathcal{H}_p(\mathbf{X}^p, \mathbf{W}) \end{aligned} \tag{3.4}$$

Definition 3.14. *The homophily levels* $(\mathcal{C}_1, \ldots, \mathcal{C}_p)$ *together form a homophily profile.*

The individual homophily levels in a homophily profile $(\mathcal{C}_1, \ldots, \mathcal{C}_p)$ are either scalars[1] (universal homophily measurement function), vectors (categorical homophily measurement function), according to the number of distinct elements in the respective feature vector. The ensemble homophily profile $(\mathcal{C}_1, \ldots, \mathcal{C}_p)$ itself is a series of scalars, vectors or a series of scalars and vectors combined.

Definition 3.15. *An ensemble homophily measurement function is normalized if all the corresponding homophily measurement functions that are its elements are normalized.*

Definition 3.15 requires that all of the values in a homophily profile are in the $[-1, 1]$ interval. This means that different types of homophilies in two networks that have the same generic feature vectors are comparable for the same variables. In the algorithms introduced later, this comparison is done within networks. However, the different homophily values across variables are incomparable due to the fact that the functions themselves might differ.

Definition 3.16. *The function* $\mathcal{E}(\mathcal{H}_1(\mathbf{X}^1, \mathbf{W}), \ldots, \mathcal{H}_p(\mathbf{X}^p, \mathbf{W}))$ *is an unitary ensemble homophily measurement function if the encompassed homophily measurement functions are all the same.*

The concept of unitary ensemble homophily measurement functions put down in Definition 3.16 can be demonstrated with a simple example. Let us take a network of co-working relationships where the vertices are employees and the links are the working relations. Age, income and job tenure of the employees are known (these are generic vertex features) and homophily can be measured regarding these variables with Moran's I. In this case the set of Moran's I functions is an unitary ensemble homophily measurement function. If one of the functions that measure feature specific homophily would have been changed to an other homophily measurement function, the set of functions would not be anymore a unitary homophily measurement function.

[1] In the previous sections scalar valued homophily was denoted by c, here to have a unified notation system I denote scalar and vector homophily levels alike with $\mathcal{C}$.

Chapter 4: Univariate homophily rearrangement algorithms

This chapter introduces algorithms that generate networks that show a given level of homophily. The common property of these algorithms is that they only deal with homophily in one single generic feature of the vertices. Topology of the networks is unchanged, and the initial distribution of the feature is also untouched by the algorithms. Only the assignment of the feature values to vertices is manipulated by the introduced procedures. In each section I give a formal pseudo code of the algorithms and describe their mechanics in detail. The algorithms are implemented for universal and categorical homophily measurement functions that satisfy the normalization criteria described in Sections 3.1 and 3.2 of Chapter 3.

The remaining sections of this chapter present the univariate homophily rearrangement algorithms. The most simple heuristic homophily rearrangement algorithms are introduced in Section 4.1. More stringent variants of these heuristic algorithms are discussed in Section 4.2. Greedy homophily rearrangement algorithms are described in Section 4.3. All of the sections include pseudo-codes which summarize the respective algorithms.

4.1 Heuristic homophily rearrangement algorithm

Heuristic homophily rearrangement algorithms are the most simple types of homophily rearrangement algorithms introduced. The basic idea behind them is that randomly switching the feature values between a pair of vertices might increase or decrease the level of homophily. In addition, the exchange of values does not change the topology (generic vertex properties are unchanged) and the distribution of the generic vertex feature itself is unchanged. The genuine key invention is that with a sufficient number of proper random value switches a target level of homophily or heterophily is achievable. The al-

gorithm design is dependent on the chosen homophily measurement function. Because of this there are two variants of the heuristic homophily rearrangement algorithm. The version which uses universal homophily measurement functions is discussed in Subsection 4.1.1 and the one which considers categorical homophily measurement is introduced in Subsection 4.1.2.

4.1.1 Universal homophily measurement function

The heuristic homophily rearrangement algorithm on a single feature is summarized with pseudo code by Algorithm 1. The algorithm needs three inputs: a prescribed level of homophily denoted by ϕ, a vector of the feature that is behind homophily $\mathbf{x}$ and an adjacency matrix $\mathbf{W}$. The adjacency matrix can describe both binary, weighted, directed or asymmetric relationships among the vertices. The feature vector can be continuous, binary or ordinal. The working of the algorithm can be summarized as follows.

The parameter N is defined as the number of elements in feature vector $\mathbf{x}$. Essentially this equals the number of vertices in the system. The initial characteristic solution time is declared to be zero – put it simply, this is stating that $t = 0$. The initial level of homophily, here noted by c, is calculated from $\mathbf{x}$ and $\mathbf{W}$ with the normalized universal homophily measurement function $\mathcal{H}_U(\mathbf{x}, \mathbf{W})$ during the initialization. Through its run the algorithm always uses the same normalized universal homophily measurement function which gives a scalar value of homophily. The time specific homophily level ω_t equals c when the algorithm starts.

The iterative process stops when ϕ is positive and the level of homophily is higher than ϕ, or when ϕ is negative and the level of homophily is below ϕ. Importantly the logical statements connected by the OR operator cannot be true at the same time, because ϕ cannot be positive and negative. This practically means that either $\phi \geq c$ or $\phi \leq c$ is the active controlling logical block statement – they cannot be active and binding at the same time. If the condition is not satisfied, an iterative process starts until it becomes a true logical statement:

1. As a first step the characteristic solution time is increased by one. The temporary feature vector $\widetilde{\mathbf{x}}$ is set to be equal with the feature vector $\mathbf{x}$. Two random integers i and j are drawn from the $[1, N]$ interval. This is expressed in the pseudo-code by choosing i and j from a discrete uniform distribution.

2. The steps enlisted here only happen when the i^{th} and j^{th} elements of the feature vector $\mathbf{x}$ do not equal – this also ensures that different vertices were selected. The i^{th} element of the feature vector $\mathbf{x}$ is assigned in place of the j^{th} element of the temporary vector $\widetilde{\mathbf{x}}$. Similarly, the j^{th} element of the feature vector $\mathbf{x}$ is assigned to be the i^{th} element of the temporary vector $\widetilde{\mathbf{x}}$. Based on the temporary feature vector $\widetilde{\mathbf{x}}$ the temporary homophily level can be calculated with the homophily measure function – this is described by the assignment $\widetilde{c} = \mathcal{H}_U(\widetilde{\mathbf{x}}, \mathbf{W})$.

3. The following steps presume that the previous condition, see step 2, was satisfied. If it was not satisfied a new pair of vertices is selected. In addition, it has to be true that the preset homophily is positive and the previously assigned $\widetilde{c}$ is higher than c, or that the preset homophily is negative and $\widetilde{c}$ is lower than c. If the condition is satisfied, the i^{th} element of the temporary feature vector $\widetilde{\mathbf{x}}$ is assigned in place of the i^{th} element of the feature vector $\mathbf{x}$. In the same way, the j^{th} element of the temporary feature vector $\widetilde{\mathbf{x}}$ is defined to be the j^{th} element of the feature vector $\mathbf{x}$. The level of homophily is updated with the temporary homophily value – $c = \widetilde{c}$. Essentially, the two elements of the original feature vector are swapped and the new homophily level is calculated form the transformed feature vector and the original weight matrix. The swap only happens permanently when the distance from the objective homophily value is decreased.

4. In every step of the iteration the time specific homophily is set to be the current level of homophily unconditionally.

There are a few observations and points that have to be clarified about the heuristic homophily rearrangement algorithm.

- The homophily measurement function is not specified in the pseudo code, technically the algorithm might be implemented with different universal normalized homophily measure functions.

- Convergence towards the prescribed homophily is stochastic, which means that for multiple runs the algorithm might result in fundamentally different networks. This phenomenon and its consequences are investigated in Section 7.1.2 of Chapter 7.

- The feature value switching condition might be relaxed, which means that those feature value changes that result in the same homophily or heterophily are allowed. Relaxation of the switching conditions results in flat regions of the homophily level – this phenomenon is show in Section 7.1.2 of Chapter 7.

- It is self-evident that the proposed algorithms might never reach a target level of homophily or heterophily if initial distribution of the generic vertex feature satisfies certain conditions.

An R implementation of this algorithm (with Moran's I as a universal homophily measure) is attached in Appendix E as Script E.1.

Data: Target homophily level ϕ, a single feature vector x and and adjacency matrix describing the weighted edges in the network denoted by $\mathbf{W}$.
Result: Network with adjacency matrix $\mathbf{W}$, feature vector $\mathbf{x}$ where the homophily in absolute terms is at least ϕ.

$N \Leftarrow |\mathbf{x}|$
$t \Leftarrow 0$
$c \Leftarrow \mathcal{H}_U(\mathbf{x}, \mathbf{W})$
$\omega_t \Leftarrow c$
while $(\phi \geq c$ **and** $\phi > 0)$ **or** $(\phi \leq c$ **and** $\phi < 0)$ **do**
  $t \Leftarrow t + 1$
  $\widetilde{\mathbf{x}} \Leftarrow \mathbf{x}$
  $i \sim U([1, N])$
  $j \sim U([1, N])$
  if $\widetilde{\mathbf{x}}_i \neq \widetilde{\mathbf{x}}_j$ **then**
    $\widetilde{\mathbf{x}}_i \Leftarrow \mathbf{x}_j$
    $\widetilde{\mathbf{x}}_j \Leftarrow \mathbf{x}_i$
    $\widetilde{c} \Leftarrow \mathcal{H}_U(\widetilde{\mathbf{x}}, \mathbf{W})$
    if $(\widetilde{c} > c$ **and** $\phi > 0)$ **or** $(\widetilde{c} < c$ **and** $\phi < 0)$ **then**
      $\mathbf{x}_i \Leftarrow \widetilde{\mathbf{x}}_i$
      $\mathbf{x}_j \Leftarrow \widetilde{\mathbf{x}}_j$
      $c \Leftarrow \widetilde{c}$
    end
  end
  $\omega_t \Leftarrow c$
end

Algorithm 1: Pseudo code of the heuristic homophily rearrangement algorithm for a single non-categorical vertex feature

4.1.2 Categorical homophily measurement function

The variant of the heuristic homophily rearrangement algorithm which can take categorical variables as an input is described by pseudo code of Algorithm 2. The running of this algorithm needs the following: a categorical generic vertex feature (like religion or race) denoted by $\mathbf{x}$, a vector describing the group specific target homophily levels Φ, and an adjacency matrix marked by $\mathbf{W}$. This algorithm requires that the homophily target vector satisfies one of the conditions described by Inequalities (4.1) and (4.2).

$$\Phi_m > 0, \quad \forall\, m \in 1, \ldots, M \tag{4.1}$$

$$\Phi_m < 0, \quad \forall\, m \in 1, \ldots, M \tag{4.2}$$

The m index notifies the number of the category in the feature vector. Essentially, the conditions mean that either all of the group specific components of the homophily target value vector are positive or all of them are negative. This also means that for all of the groups we either have homophily or heterophily. The algorithm in addition to the above mentioned requirements needs a normalized categorical homophily measurement to be preset, in the pseudo code it is the expression $\mathcal{H}_C(\mathbf{x}, \mathbf{W})$.

Initialization of the algorithm can be summarized as follows. The number of elements in the categorical feature vector is counted and the number of elements (which equals to the number of vertices) is assigned to N. The initial time running time is set to be zero. For each of the groups the groups specific homophily levels are calculated and stored in $\mathcal{C}$. The time specific homophily at time t are stored in Ω_t.

After the initialization, an iterative process starts which is controlled by a block statement related to the target homophily vectors. The iterative process only stops when convergence to the preset level of homophily happens. Vector $\mathbf{0}$ has elements equal to the number of groups in feature vector $\mathbf{x}$, and all of its elements are zeros. One of the possible loop breaking outcomes is when elements of the target homophily vector were higher than zero and the group specific homophily levels are all higher than the target homophily vector's values. Fundamentally this means that $\Phi \succeq \mathcal{C}$ and $\Phi \succ \mathbf{0}$ is true at the same time. The another outcome which stops the iterative process is happening when all of the groups specific homophily values are lower than the target homophily vectors elements pairwise and the elements of the target homophily vector are all negative. This means that we know that $\Phi \preceq \mathcal{C}$ and also that $\Phi \prec \mathbf{0}$.

1. First the characteristic solution time of the problem is increased to be the next integer in order. The feature vector $\mathbf{x}$ is assigned to be the temporary feature vector $\widetilde{\mathbf{x}}$. A pair of random integers, respectively i and j, are drawn from the interval $[1, N]$.

2. The subsequent operations only take place if it is true that the i^{th} and j^{th} elements of the generic feature vector do not have the same value. Just as before, this also means that one chose different vertices of the network. The temporary generic feature vectors j^{th} element is replaced with the i^{th} element of the feature vector. In an analogous way, the i^{th} element of the temporary feature vector is replaced with the j^{th} element of the feature vector. The resulting temporary feature vector $\widetilde{\mathbf{x}}$ is used for calculating the temporary homophily level vector $\widetilde{\mathcal{C}}$.

3. The following procedures takes place in two scenarios. The first possibility is that the target homophily was positive for every group and for all of the groups the homophily increased. The second is that the prescribed homophily was negative for all of the groups and the group specific homophily values all decreased . If one of these conditions was satisfied the following happens: the i^{th} element of the feature vector $\mathbf{x}$ is replaced with the i^{th} element of the temporary feature vector $\widetilde{\mathbf{x}}$. Furthermore, the j^{th} element of the feature vector $\mathbf{x}$ is replaced with the j^{th} element of the temporary feature vector $\widetilde{\mathbf{x}}$. In addition, the homophily criteria $\mathcal{C}$ is replaced with the temporary homophily criteria $\widetilde{\mathcal{C}}$.

4. Irrespective of the change in temporary homophily values and on the selected vertices, the time specific homophily vector Ω_t is set to be equal with the the homophily criteria vector $\mathcal{C}$.

Just like in the case of the heuristic universal homophily rearrangement algorithm, the switching condition of the categorical variant can be relaxed. In the relaxed model, the group specific homophily levels might stay constant in iterative steps that result in a permanent feature value exchange. This phenomenon is demonstrated by simulation results in Section 7.1.2 of Chapter 7. It is worth emphasizing that in each time period a time specific homophily level vector is generated. At the end of the iterative process one has $t+1$ time specific homophily level vectors which allow for comparing the time dependent evolution of group specific homophily levels. For example, the group specific homophily levels can be plotted as a function of time. Moreover, other group specific topological features can be monitored during the iterative process – such as mean betweenness centrality, coreness or clustering coefficient in the group.

As simulations are essential parts of the thesis, an R implementation of this algorithm (with the homophily index as a categorical homophily measure) is enclosed as Script E.2 in Appendix E.

Data: Homophily target vector Φ, a single feature vector $\mathbf{x}$ and adjacency matrix describing the weighted edges in the network denoted by $\mathbf{W}$.

Result: Network with adjacency matrix $\mathbf{W}$ and feature vector $\mathbf{x}$ where the group specific homophily values are greater than the corresponding elements of the vector Φ.

$N \Leftarrow |\mathbf{x}|$
$t \Leftarrow 0$
$\mathcal{C} \Leftarrow \mathcal{H}_C(\mathbf{x}, \mathbf{W})$
$\Omega_t \Leftarrow \mathcal{C}$
while $(\Phi \succeq \mathcal{C}$ **and** $\Phi \succ \mathbf{0})$ **or** $(\Phi \preceq \mathcal{C}$ **and** $\Phi \prec \mathbf{0})$ **do**
  $t \Leftarrow t + 1$
  $\widetilde{\mathbf{x}} \Leftarrow \mathbf{x}$
  $i \sim U([1, N])$
  $j \sim U([1, N])$
  if $\widetilde{\mathbf{x}}_i \neq \widetilde{\mathbf{x}}_j$ **then**
    $\widetilde{\mathbf{x}}_i \Leftarrow \mathbf{x}_j$
    $\widetilde{\mathbf{x}}_j \Leftarrow \mathbf{x}_i$
    $\widetilde{\mathcal{C}} \Leftarrow \mathcal{H}_C(\widetilde{\mathbf{x}}, \mathbf{W})$
    if $(\widetilde{\mathcal{C}} \succ \mathcal{C}$ **and** $\Phi \succ \mathbf{0})$ **or** $(\widetilde{\mathcal{C}} \prec \mathcal{C}$ **and** $\Phi \prec \mathbf{0})$ **then**
      $\mathbf{x}_i \Leftarrow \widetilde{\mathbf{x}}_i$
      $\mathbf{x}_j \Leftarrow \widetilde{\mathbf{x}}_j$
      $\mathcal{C} \Leftarrow \widetilde{\mathcal{C}}$
    end
  end
  $\Omega_t \Leftarrow \mathcal{C}$
end

Algorithm 2: Pseudo code of the heuristic homophily rearrangement algorithm for a single categorical generic vertex feature

4.2 Heuristic homophily rearrangement algorithm with bag of indices

The heuristic homophily rearrangement algorithms with bag of indices are effective iterative step number restricted versions of the homophily rearrangement algorithms introduced in Section 4.1. First, an effective iterative step results in a homophily/heterophily level that is closer to the target. Second, the restriction means that the number of effective steps is limited. Simply, the presented algorithms have a limit on this effective step number. The algorithms presented in this section rearrange a single vertex feature to obtain a target level of homophily. This target level can be either a scalar value or a vector of group specific homophily values if the vertex feature is categorical. The variant of the algorithm which is able to deal with count, continuous and binary variables is discussed in Subsection 4.2.1, while the one that can deal with categorical ones is described with pseudo-code in Subsection 4.2.2.

4.2.1 Universal homophily measurement function

The heuristic homophily rearrangement algorithm might be relaxed (virtually the feature value exchanges can take place that do not decrease homophily). Importantly it follows that mutual pairwise exchanges might take place between nodes that do not get homophily closer to the target level. This means that restricting vertices to participate in more than one value exchange during the iterative process might help the convergence – as stated above this might be especially true when the switching condition is relaxed to be an inequality.

The heuristic homophily rearrangement algorithm with a bag of indices for an universal homophily measurement function is described with pseudo-code by Algorithm 3. The algorithm needs an adjacency matrix which represents the network, a single generic vertex feature which is binary, count or continuous, and a normalized universal homophily measurement function. The homophily rearrangement algorithm is initialized by counting the number of vertices, which is practically the number of elements in the generic feature vector $\mathbf{x}$. This number is assigned to be the scalar N. After this, the bag of indices is defined to be the set of integers starting from 1 to N inclusive – this set is denoted by $\mathcal{B}$. The time ticker is set to be zero, and the universal level of homophily c is calculated from the feature vector and the adjacency matrix. Finally, the time specific homophily

level is equalized with the level of homophily. After the initialization the algorithm starts an iterative process, in the next few paragraphs I synthesize this iterative mechanism.

1. The iterative process stops when homophily is at least the target homophily ϕ or when heterophily is at least below the target homophily. When these conditions are unsatisfied the following steps take places. The time ticker is incremented by one, the temporary feature vector $\widetilde{\mathbf{x}}$ is set to be equal the feature vector $\mathbf{x}$. Two random integers i and j are chosen from the bag of indices. The probability of choosing a given index is the same for every index – this is why the probability distribution over $\mathcal{B}$ is discrete uniform.

2. The process takes the following steps only if the respective elements of the feature vector do not equal with each other. The i^{th} element of the temporary feature vector is set to be the j^{th} element of the generic vertex feature. In a similar manner, the j^{th} element of the temporary feature vector is set to be the i^{th} element of the feature vector. Essentially, these two steps mean the feature exchange. With the use of the temporary feature vector a temporary measure of universal homophily is calculable, here denoted by $\widetilde{c}$.

3. The finalization of the feature value exchange takes place on two specific occasions. First, if the temporary homophily criterion is higher than the permanent homophily, when the target is homophily. This simply means that $\widetilde{c} > c$ and $\phi > 0$ are satisfied at the same time. Second, if the temporary homophily criterion is lower than the permanent homophily, while the target was heterophily. This connotes that $\widetilde{c} < c$ and $\phi < 0$ are true at the same time. If the above mentioned are satisfied, the following steps finalize the feature value exchange: the i^{th} element of the feature vector is assigned to be the i^{th} element of the temporary vertex feature. In a similar manner, the j^{th} element of the temporary feature vector is assigned to be the i^{th} element of the feature vector. The universal homophily level is c, assigned to be the temporary universal homophily level $\widetilde{c}$. The indices i and j are removed from the bag of indices – the corresponding vertices cannot participate in further feature value switchings.

4. Independent of any condition on homophily level improvement, the time dependent homophily level is set to be the homophily level.

The heuristic nature of the algorithm postulates that it might never converge to the target homophily or heterophily level. Moreover, it might *end up being stalled in an infinite*

while loop – when the bag of indices is empty and the homophily target value condition is not satisfied. The pseudo-code can be augmented with a termination condition to stop infinite while loops. A simple R implementation of the algorithm is enclosed as Script E.3 in Appendix E.

Data: Target homophily value ϕ, a single feature vector $\mathbf{x}$ and adjacency matrix describing the weighted edges in the network denoted by $\mathbf{W}$.

Result: Network with adjacency matrix $\mathbf{W}$ and feature vector $\mathbf{x}$ where the homophily in absolute terms is at least ϕ.

$N \Leftarrow |\mathbf{x}|$
$\mathcal{B} \Leftarrow \{1, \ldots, N\}$
$t \Leftarrow 0$
$c \Leftarrow \mathcal{H}_U(\mathbf{x}, \mathbf{W})$
$\omega_t \Leftarrow c$
while $(\phi \geq c$ **and** $\phi > 0)$ **or** $(\phi \leq c$ **and** $\phi < 0)$ **do**
  $t \Leftarrow t + 1$
  $\widetilde{\mathbf{x}} \Leftarrow \mathbf{x}$
  $i \sim U(\mathcal{B})$
  $j \sim U(\mathcal{B})$
  if $\widetilde{\mathbf{x}}_i \neq \widetilde{\mathbf{x}}_j$ **then**
    $\widetilde{\mathbf{x}}_i \Leftarrow \mathbf{x}_j$
    $\widetilde{\mathbf{x}}_j \Leftarrow \mathbf{x}_i$
    $\widetilde{c} \Leftarrow \mathcal{H}_U(\widetilde{\mathbf{x}}, \mathbf{W})$
    if $(\widetilde{c} > c$ **and** $\phi > 0)$ **or** $(\widetilde{c} < c$ **and** $\phi < 0)$ **then**
      $\mathbf{x}_i \Leftarrow \widetilde{\mathbf{x}}_i$
      $\mathbf{x}_j \Leftarrow \widetilde{\mathbf{x}}_j$
      $c \Leftarrow \widetilde{c}$
      $\mathcal{B} \Leftarrow \mathcal{B} \setminus \{i, j\}$
    end
  end
  $\omega_t \Leftarrow c$
end

Algorithm 3: Pseudo code of the heuristic homophily rearrangement algorithm with bag of indices for a single generic vertex feature

4.2.2 Categorical homophily measurement function

The categorical heuristic homophily rearrangement algorithm with a bag of indices is a restricted version of the heuristic homophily rearrangement algorithm for categorical variables. The individual vertices can only participate in one effect feature value exchange. The algorithm needs a network with a single binary, categorical or ordinal generic vertex feature $\mathbf{x}$. The homophily measurement function needs to be normalized and categorical. In addition, a vectorial target of group specific homophily values has to be set. This Φ vector must have only positive or negative components as group specific target values.

A pseudo-code summary of the algorithm is included as Algorithm 4. To start the algorithm one has to declare the number of vertices to be N, just as in Algorithms 1, 2 and 3. The bag indices is set to be the series of integers between 1 and N. The time index is set to be zero, and the level of homophily is quantified with the categorical homophily measurement function specified for the algorithm. The time specific vector of homophily levels is equalized with the level of homophily in the network regarding the generic vertex feature $\mathbf{x}$. The iterative process that starts after initialization is similar to the one implemented in Algorithm 2. In the consequential paragraphs I summarize this process.

1. The iterative process stops on two occasions. First, when all the groups specific homophily levels are above the target homophily levels – formally, when ($\Phi \succeq \mathcal{C}$ and $\Phi \succ \mathbf{0}$). Second, when all the group specific heterophily levels are at the heterophily threshold or below. To put simply, it holds that $\Phi \preceq \mathcal{C}$ and $\Phi \prec \mathbf{0}$. Again, intuitively only one of the controlling block statements is active – the goal is either homophily for every group or heterophily. While one of the controlling block statements is true, the time ticker is increased by one. The temporary feature vector is set to be equal to the single generic vertex feature. Two random vertex indices i and j are chosen from $\mathcal{B}$.

2. The iterative process envoys by exchanging feature values if the feature values corresponding to the indices do not equal. The i^{th} element of the temporary feature vector overwritten be the j^{th} element of $\mathbf{x}$. Likewise, the j^{th} element of the temporary feature vector is overwritten by the i^{th} element of $\mathbf{x}$. The obtained temporary feature vector is used for calculating the temporary categorical homophily level on the network.

Data: Homophily target vector Φ, a single categorical vertex feature vector $\mathbf{x}$ and adjacency matrix describing the weighted edges in the network denoted by $\mathbf{W}$.

Result: Network with adjacency matrix $\mathbf{W}$ and feature vector $\mathbf{x}$ where the group specific homophily values are greater than the corresponding elements of vector Φ.

$N \Leftarrow |\mathbf{x}|$
$\mathcal{B} \Leftarrow \{1, \ldots, N\}$
$t \Leftarrow 0$
$\mathcal{C} \Leftarrow \mathcal{H}_C(\mathbf{x}, \mathbf{W})$
$\Omega_t \Leftarrow \mathcal{C}$
while $(\Phi \succeq \mathcal{C}$ **and** $\Phi \succ \mathbf{0})$ **or** $(\Phi \preceq \mathcal{C}$ **and** $\Phi \prec \mathbf{0})$ **do**
    $t \Leftarrow t + 1$
    $\widetilde{\mathbf{x}} \Leftarrow \mathbf{x}$
    $i \sim U(\mathcal{B})$
    $j \sim U(\mathcal{B})$
    if $\widetilde{\mathbf{x}}_i \neq \widetilde{\mathbf{x}}_j$ **then**
        $\widetilde{\mathbf{x}}_i \Leftarrow \mathbf{x}_j$
        $\widetilde{\mathbf{x}}_j \Leftarrow \mathbf{x}_i$
        $\widetilde{\mathcal{C}} \Leftarrow \mathcal{H}_C(\widetilde{\mathbf{x}}, \mathbf{W})$
        if $(\widetilde{\mathcal{C}} \succ \mathcal{C}$ **and** $\Phi \succ \mathbf{0})$ **or** $(\widetilde{\mathcal{C}} \prec \mathcal{C}$ **and** $\Phi \prec \mathbf{0})$ **then**
            $\mathbf{x}_i \Leftarrow \widetilde{\mathbf{x}}_i$
            $\mathbf{x}_j \Leftarrow \widetilde{\mathbf{x}}_j$
            $\mathcal{C} \Leftarrow \widetilde{\mathcal{C}}$
            $\mathcal{B} \Leftarrow \mathcal{B} \setminus \{i, j\}$
        end
    end
    $\Omega_t \Leftarrow \mathcal{C}$
end

Algorithm 4: Pseudo code of the heuristic homophily rearrangement algorithm with bag of indices for a single categorical generic vertex feature

3. Similar to other algorithms implemented previously in my thesis the temporary feature exchange is becoming permanent in two cases. One case is when the temporary homophily profile is element-wise greater than the elements of the actual homophily profile, and the target was homophily for each group. Another case is when the temporary homophily profile is element-wise below the actual homophily profile and the target vector described group wise heterophily. In these two cases, the i^{th} element of $\mathbf{x}$ becomes the i^{th} element of $\widetilde{\mathbf{x}}$. Correspondingly, the j^{th} element of $\mathbf{x}$ becomes the j^{th} element of $\widetilde{\mathbf{x}}$. The vector that describes the homophily level is replaced by the temporary homophily level. Finally, the indices i and j are removed from the bag of indices.

4. The time specific homophily vector Ω_t is equalized with the homophily vector.

This algorithm only differs from Algorithm 2 in that the indices are selected from the bag of indices. Importantly, the exchange conditions are relaxable and like the heuristic homophily rearrangement algorithm this one is also prone to infinite loops. The iterative assignment of time specific homophily levels once again allows for time-dependent tracking of homophily levels. An R implementation of the algorithm is enclosed as Script E.4 in Appendix E.

4.3 Greedy homophily rearrangement algorithm

The heuristic homophily rearrangement algorithm presented in Section 4.1 and its restricted variant, the heuristic homophily rearrangement algorithm with bag of indices, discussed in Section 4.2 select the vertices randomly in the feature switching process. This results in feature value changes that do not move the level of homophily towards the target homophily level. In each step the level of homophily has to be measured – and this step-wise measurement is computationally costly. Because of this, in large complex networks the convergence to a prescribed level of homophily or heterophily takes a large number of iterative steps. In order to make switches of feature values that move the target homophily level in the right direction in each iterative step, one can come up with decision rules about how to choose the right vertices in the network. In addition, these decision rules might result in changes in the target value that are larger in absolute terms on average than the changes that would result from the random switches of the feature values. In order to treat these algorithms in a generalized way I provide certain definitions at the beginning of this section.

On the basis of the set of criteria laid down by Cormen et al. (2009), the design of Algorithms 5 and 6 discussed in the section is greedy. First, both algorithms have a well defined *candidate set of vertex pairs* for the feature exchanges (the sub optimization problems that have to be solved) – either based on the feature mean or a randomly chosen categorical vertex feature value. Second, the algorithms both use a *selection heuristic* which ensures that proper vertex pairs are chosen in the iterations in order to move the objective value in the right direction. Third, after selecting a pair of vertices, the algorithms check whether the selected pair of indices can contribute to the homophily rearrangement in a way that there is a *feasibility function* element in the designs. Fourth, in each case there is an *objective function* included in the algorithms, namely, the level of homophily or heterophily in the system. Fifth, the algorithms have a *stopping criteria* which indicates that a final solution to the homophily rearrangement problem is achieved. The co-existence of these 5 elements together signify that these algorithms are greedy indeed.

Definition 4.1. *Let us have a network with a single generic vertex feature represented by* $\mathrm{G}(\mathbf{x}, \mathbf{W})$*. The network* $\mathrm{G}(\mathbf{x}, \mathbf{W})$ *has* N *vertices, and the set* V *represents the corresponding vertex indices of the network from 1 to* N*. The first order neighboring vertices of vertex* i *are defined as:* $V_i = \{j \in V : \mathbf{W}_{i,j} > 0\}$.

The first order neighbors concept discussed in Definition 4.1 is needed for defining a subnetwork of the original network which contains the neighbors of a given vertex. Only those vertices are included in the neighborhood of vertex i which have a relationship with i – importantly these relationships are possibly weighted, that is why I used an inequality. The use of an inequality makes my neighborhood definition different from the one that Fagiolo et al. (2007) introduced, who only consider unweighted neighbors and use an equality ($\mathbf{W}_{i,j} = 0$) instead of an inequality. Moreover, Fagiolo et al. (2007) require that $\mathbf{W}_{i,j} = \mathbf{W}_{j,i}$ which is always satisfied in the setting of my thesis, because the weighted adjacency matrix is defined to be symmetric in my thesis.

Definition 4.2. *The vertex* i *induced star subnetwork of* $\mathrm{G}(\mathbf{x}, \mathbf{W})$ *is a network which includes* i *and its first order neighboring vertices. In the vertex* i *induced star subnetwork edges only exist between* i *and its neighboring vertices. The vertex* i *induced star subnetwork of* $\mathrm{G}(\mathbf{x}, \mathbf{W})$ *is connoted by* $\mathrm{G}_i^*(\mathbf{x}^*, \mathbf{W}^*)$.

It should be emphasized that Definition 4.2 does not allow the existence of edges among vertex i's neighbors. This restriction is needed because the similarity of i's neighboring vertices is not in focus when one investigates the dissimilarity of i itself from its neighbors.

We know that exchanging feature values between vertices that are starkly dissimilar from their neighbors and have different vertex feature values from each other can increase homophily. Similarly, if one is able to change the feature values between vertices that are similar to their neighbors regarding the generic vertex feature and different from each other the heterophily in the network can be increased in an effective greedy way.

Definition 4.3. *The network denoted by* $\mathrm{G}(\mathbf{x}, \mathbf{W})$ *has the single feature* $\mathbf{x}$*, and in this network* $\mathrm{G}_i^*(\mathbf{x}^*, \mathbf{W}^*)$ *is a vertex* i *induced star subnetwork. In this subnetwork the scalar* d_i^* *is defined to be the average feature dissimilarity of vertex* i *from its neighbors.*

The average feature dissimilarity in Definition 4.3 is a simple concept, the dissimilarity of a vertex from its neighbors can be quantified by averaging out the pairwise dissimilarities between the central vertex and the neighbors. The dissimilarity measure that is averaged out has to be the same for all considered relationships. Furthermore, when choosing such measure one has to take into account that different types of generic vertex features (for example, the ordinal and continuous variables) might need different types of dissimilarity measures (Deza & Deza, 2009).

Definition 4.4. *The degree corrected average feature dissimilarity of node* i *is denoted by* Δ_i *and equals to* $\deg(i) \cdot d_i$*, where* $\deg(i)$ *is the degree of vertex* v *in the original network* $\mathrm{G}(\mathbf{x}, \mathbf{W})$*, and* d_i^* *is the average feature dissimilarity of node* i*.*

The measure set down in Definition 4.4 is needed because the average dissimilarity of a vertex does not take into account the fact that a node that is strongly dissimilar from its neighbors can have a low degree. The degree corrected average dissimilarity takes into account the connectedness of the inspected vertex – it is practically the sum of pairwise dissimilarity measures in the i induced star subnetwork.

4.3.1 Universal homophily measurement function

The greedy homophily rearrangement algorithm which uses universal homophily measurement functions is included as Algorithm 5. The algorithm needs a specified target homophily level, a network with adjacency matrix $\mathbf{W}$ which has a single generic vertex feature $\mathbf{x}$ which is either binary, continuous or count. The universal homophily measurement function has to be preset and the universal vertex selection algorithm $\mathcal{S}_U$ has to be predefined – later in this section I show a very specific vertex selection algorithm that I use for homophily rearrangement algorithm design.

1. During the initialization the control flag Θ is set to be zero, likewise the time ticker t is equalized with zero. The initial level of homophily is measured regarding the generic vertex feature, and the time specific homophily level ω_t is assigned to be the level of homophily. After these steps an iterative process starts.

2. The iterative process stops on three occasions: First when the achieved homophily level is at least the target homophily ϕ. Second, the target was heterophily and the heterophily level is at least below ϕ. Third, the iterative process stops because it is halted by an exit flag equal to one – this is connected to the previous conditions by an AND operator.

3. During the run of the iterations the time ticker value is incremented by one. The temporary feature vector is equalized with the feature vector. The universal vertex index selection algorithm based on the target homophily level, the feature vector and the adjacency matrix chooses a pair of indices for the feature value exchange. A simplistic greedy universal index selection method is described by Algorithm 11 in Appendix B. It works as follows:

   (a) The scalar N is set to be the number of vertices in the system. The set V describes the set of vertex indices in the network. If the original target of the homophily rearrangement algorithm calling the index selection algorithm is homophily the above the mean ($\mathcal{X}^{\uparrow}$) and the below the mean ($\mathcal{X}^{\downarrow}$) control values are symbolically set to to be equal to $-\infty$. In a similar manner, if the target of the greedy homophily rearrangement algorithm the control values both equal to ∞. After this an iterative process goes through the vertices in the system.

      i. The output index i is equalized with index v and the above the mean control value is set to be the degree corrected average feature dissimilarity of node v if three conditions hold at the same time. First, the generic feature value is below the vertex feature's mean. Second, the target of the homophily rearrangement was obtaining homophily. Third, the degree corrected average feature dissimilarity of node v is higher than the above the mean control value.

      ii. Similarly, the index i is overwritten by v and the above the mean control value is set as the degree corrected average feature dissimilarity if three conditions are true. First, the $\mathbf{x}_v$ value is above the mean of $\mathbf{x}$. Second, the target was a network which shows heterophily regarding the variable.

Third, the degree corrected average feature dissimilarity of node v is lower than the above the mean control value.

iii. Alike, the returning index j is set to be v, and the below the mean control value is substituted with the degree corrected average feature dissimilarity of v if three logical statements hold. First, the feature value is below the mean of the feature. Second, the target network shows homophily regarding $\mathbf{x}$. Third, the d_v^* value is above the current below the mean control value.

iv. Finally, index j is set to be v and the below the mean control value is replaced by the degree corrected average dissimilarity of vertex v if three conditions stand at the same time. First, the vertex has a feature value that is below the feature's mean. Second, the target network that one wants to achieve shows heterophily regarding $\mathbf{x}$. Third, the degree corrected average feature dissimilarity of v is below the current below the mean control value.

(b) The universal vertex selection algorithm returns the indices i and j as the vertices to be chosen next for the feature value rearrangement.

4. Based on a pair of indices that were returned by the universal vertex selection algorithm, the replacements in the temporary feature vector take place. The i^{th} element of the temporary feature vector is replaced by the j^{th} element of the actual feature vector. Likewise, the j^{th} element in the temporary vector is replaced by the i^{th} element of the actual feature vector. With a temporary feature vector a temporary homophily criteria (level) is calculated and assigned to be $\widetilde{c}$.

5. If the target was homophily ($\phi > 0$), and the homophily level is increased by the temporary exchange, the temporary feature vectors respective elements replace the elements in the feature vector and the temporary homophily criteria is replaced by the homophily criteria. In addition, if the target was heterophily ($\phi < 0$) and the homophily decreased (heterophily increased) the feature value exchange becomes permanent and the temporary homophily criteria replaces the homophily criteria. Otherwise the exit flag is set to be one and the iterative process will terminate in the next iteration.

6. The homophily is assigned to be the time dependent homophily level unconditionally in each iterative step.

Data: Target homophily level ϕ, a single feature vector x and adjacency matrix describing the weighted edges in the network denoted by $\mathbf{W}$.

Result: Network with adjacency matrix $\mathbf{W}$ and feature vector $\mathbf{x}$ where the homophily in absolute terms is at least ϕ.

$\Theta \Leftarrow 0$
$t \Leftarrow 0$
$c \Leftarrow \mathcal{H}_U(\mathbf{x}, \mathbf{W})$
$\omega_t \Leftarrow c$
while $[(\phi \geq c$ **and** $\phi > 0)$ **and** $(\Theta = 0)]$ **or** $[(\phi \leq c$ **and** $\phi < 0)$ **and** $(\Theta = 0)]$ **do**
    $t \Leftarrow t + 1$
    $\widetilde{\mathbf{x}} \Leftarrow \mathbf{x}$
    $\{i, j\} \Leftarrow \mathcal{S}_U(\phi, \mathbf{x}, \mathbf{W})$
    $\widetilde{\mathbf{x}}_i \Leftarrow \mathbf{x}_j$
    $\widetilde{\mathbf{x}}_j \Leftarrow \mathbf{x}_i$
    $\widetilde{c} \Leftarrow \mathcal{H}_U(\widetilde{\mathbf{x}}, \mathbf{W})$
    if $(\widetilde{c} > c$ **and** $\phi > 0)$ **or** $(\widetilde{c} < c$ **and** $\phi < 0)$ **then**
        $\mathbf{x}_i \Leftarrow \widetilde{\mathbf{x}}_i$
        $\mathbf{x}_j \Leftarrow \widetilde{\mathbf{x}}_j$
        $c \Leftarrow \widetilde{c}$
    else
        $\Theta \Leftarrow 1$
    end
    $\omega_t \Leftarrow c$
end

Algorithm 5: Pseudo code of the greedy homophily rearrangement algorithm for a single vertex feature

About Algorithm 5 and the auxiliary vertex selection heuristic described by Algorithm 11 in Appendix B a few important observations and notions have to be highlighted. These can be briefly summarized as follows.

If the greedy algorithm stalls in an infinite while loop, namely because the exchange of features do not move homophily in the direction of the target homophily, the algorithm is stopped by the exit flag. This is particularly intriguing, because one will have information

whether the greedy algorithm was able to tackle the homophily rearrangement problem or not – this is just a product of the greedy algorithm design.

The vertex selection heuristic (this is true for the categorical case also) implemented in the thesis is computationally complex. In each iterative step, the number of pairwise dissimilarities calculated equals to the number of links in the network. Moreover, as vertex feature values are exchanged these dissimilarities are recalculated in each iterative step in order to choose indices that move the objective value towards the target level. This means that the relative potential of a given vertex to increase or decrease homophily changes with each iterative step.

The selection heuristic chooses the vertices based on a comparison that is done for all of the individual vertices. This means that once the pairwise dissimilarities are obtained the *order of the vertex selection algorithm* equals to the number of vertices in the system. Recalculating the dissimilarities in each iterative step only for the affected vertices (the pair of nodes that participate in the exchange and their neighbors) would make the algorithm computationally cheaper. On the other hand, as a trade off all of the pairwise dissimilarities should be stored.

As it is clearly stated, the proposed selection of the vertex pair that participates in the feature exchange is partially based on whether a vertex has a feature value below or above the mean. The rational behind this is that universal homophily measurement functions take into account deviations from the feature mean explicitly. The Moran's I and the transformed Geary's C both consider deviations from mean of the feature. My thesis only considers a single index selection mechanism. However, other selection algorithms might be proposed to choose the pair of indices. It is self-evident that choosing a different vertex selection mechanism might change the simulation results presented later in the paper.

The R implementation of the main algorithm appears as Script E.5 in Appendix E. The auxiliary index selection mechanism is attached as E.6 in Appendix E. The algorithm assumes that the universal homophily measurement function is Moran's I.

4.3.2 Categorical homophily measurement function

The greedy algorithm is extendable in a way that it can rearrange a single categorical or ordinal feature in order to achieve a given vector of group specific homophily/heterophily

levels in the network – the pseudo-code sketch of the method is shown by Algorithm 6. The algorithm is defined in a generic way, this means that the categorical homophily measurement function is left unspecified. In order to choose the vertices participating in the feature value exchange in a greedy way, the algorithm needs a *categorical vertex selection algorithm*. The before mentioned categorical vertex feature selection algorithm is chosen arbitrarily. A self-developed algorithm which is suitable can be found as Algorithm 12 in Appendix B.

In order to start the homophily rearrangement one has to set the exit flag Θ to be zero. The time ticker is also zero in the beginning, the $\mathcal{C}$ homophily vector contains the initial group specific homophily values and the time specific homophily vector Ω_t inherits its value from the homophily vector. After this an iterative process of feature value exchanges starts in order to achieve the given level of homophilies. The steps of the iterations and the controlling statements of the iteration itself are subsumed as:

1. The algorithm stops when the group specific homophily values are above the specified target Φ, or when the respective heterophily values are below the elements of Φ. Furthermore, the algorithm also stops when the exit flag is not zero anymore. If the stopping conditions are not satisfied the subsequent steps take place.

2. The time tickers is increased by one. The temporary feature vector $\widetilde{\mathbf{x}}$ is replaced by the categorical feature vector $\mathbf{x}$. A categorical index selection algorithm $\mathcal{S}_C$, based on the target vector Φ, the feature vector $\mathbf{x}$ and the adjacency matrix $\mathbf{W}$ is used for choosing a pair of vertices (i and j) for the feature value exchange. This auxiliary algorithm is described by Algorithm 12 in Appendix B. Basic mechanics of the vertex selection algorithm can be summarized as:

   (a) The number of elements in the vertex feature is assigned to be N. The set V contains the vertex indices from 1 to N. The index of unique feature values is contained by the set $\mathcal{M}$, and a random feature is assigned to be the scalar m. If the target is achieving homophily for each of the groups, then the within and outside group control values (respectively $\mathcal{X}^m$ and $\mathcal{X}^{\not m}$ with my notations) are set to be $-\infty$. Otherwise, in case of a heterophily target both of them is set to be ∞.

   (b) An iterative process loops through the indices in V and for each vertex v the degree corrected average dissimilarity is calculated. Based on the target vector of the homophily rearrangement algorithm, the corresponding feature value of

the vertex and the investigated node's degree corrected average dissimilarity the following can happen:

i. The vertex index i is replaced by index v, the withing group control value $\mathcal{X}^m$ is replaced by the degree corrected average feature dissimilarity of node v, if the following three criteria hold together. First, the vertex specific feature value $\mathbf{x}_v$ equals to the randomly chosen category m. Second, the target of the rearrangement algorithm was homophily – formally it means that $\Phi \succ \mathbf{0}$.Third, the degree corrected average dissimilarity is higher than the within group specific control value.

ii. Analogously, index i is set to be index v and $\mathcal{X}^m$ is equalized with $\deg(v) \cdot d_v^*$ under the withstanding of three criteria. First, the feature value $\mathbf{x}_v$ corresponding to index v has to be equal to m. Second, target of the algorithm has to be heterophily for all of the groups. Third, the degree corrected average feature dissimilarity has to be lower than the withing group control value.

iii. In a similar manner, the returning index j is transcribed by v, and the outside the group control value is replaced by the vertex v specific degree corrected average feature dissimilarity. These replacements take place if the following three statements are all true. First, the feature value does not equal to m. Second, the target vector Φ described homophily. Third, $\deg(v)$ is greater than the outside the group specific control value $\mathcal{X}^{\not m}$

iv. Akin, output index j is substituted out by index v and the outside the group control value $\mathcal{X}^{\not m}$ is replaced by $\deg(v) \cdot d_v^*$ if three separate conditions are true. First, the index v specific feature value does not equal m. Second, the target was heterophily for each categorical groups of $\mathbf{x}$. Third, the degree corrected average feature dissimilarity of vertex v is below the outside the group control value.

(c) The vertex selection algorithm returns the index pair consisting i and j.

3. The pair of indices returned by $\mathcal{S}_C$ is used for exchanging values between the temporary and actual feature vector. The $\mathbf{x}_i$ is assigned in place of $\widetilde{\mathbf{x}}_j$, while the $\mathbf{x}_j$ value is assigned in place of $\widetilde{\mathbf{x}}_i$. After the feature value exchange took place based on the temporary feature vector a temporary homophily vector is calculated.

4. If the temporary homophily levels are above the actual ones, while the target vector Φ described homophily or when the temporal heterophily levels are below the actual

heterophily level and the target vector Φ described heterophily, additional operations take place. The i^{th} element of the temporary feature vector is assigned to be the i^{th} element of the feature vector. Likewise, the j^{th} element of the temporary feature vector is set as the j^{th} element of the feature vector. The temporary homophily vector overwrites the homophily vector. If the preconditions are not satisfied, Θ is set to be one which results in termination of the iterative process.

5. Independent from any of the above mentioned conditions the time dependent homophily levels are set to be the homophily levels.

The most novel contribution of the greedy homophily rearrangement algorithm which can deal with categorical variables is the vertex selection heuristic which chooses the pair of vertices that participate in the feature exchange. In each iterative step one of the unique values of the categorical feature are chosen to notify membership in a reference group. The degree corrected average dissimilarity values are calculated for all of the individual vertices. Later based on the degree corrected average dissimilarity levels, two vertices are chosen, one of them from those vertices that have an $\mathbf{x}$ value equal to the reference group's value, and another vertex from those that do not belong to the reference group. Importantly, this selection heuristic treats ordinal variables in a way that there is no meaning of their structured nature – only common group membership matters.

Random choice of the specific feature value that is used for reference ensures that none of the groups ends up with an abnormally high group specific homophily value. For example, if one has a categorical generic vertex feature which can take 4 different unique values and one group is always participating in the feature value exchanges than the chosen group's homophily value might increase faster than homophily regarding the other groups.

The main algorithm's R implementation is enclosed as Script E.7 in Appendix E. The vertex pair selection method's R implementation is in Appendix E as Script E.8

Data: Homophily target vector Φ, a single feature vector $\mathbf{x}$ and an adjacency matrix describing the weighted edges in the network denoted by $\mathbf{W}$.

Result: Network with adjacency matrix $\mathbf{W}$ and feature vector $\mathbf{x}$ where the group specific homophily values are greater than the corresponding elements of the vector Φ.

$\Theta \Leftarrow 0$
$t \Leftarrow 0$
$\mathcal{C} \Leftarrow \mathcal{H}_C(\mathbf{x}, \mathbf{W})$
$\Omega_t \Leftarrow \mathcal{C}$
while $[(\Phi \succeq \mathcal{C}$ **and** $\Phi \succ \mathbf{0})$ **or** $(\Phi \preceq \mathcal{C}$ **and** $\Phi \prec \mathbf{0})]$ **and** $(\Theta = 0)$ **do**
    $t \Leftarrow t + 1$
    $\widetilde{\mathbf{x}} \Leftarrow \mathbf{x}$
    $\{i, j\} \Leftarrow \mathcal{S}_C(\Phi, \mathbf{x}, \mathbf{W})$
    $\widetilde{\mathbf{x}}_i \Leftarrow \mathbf{x}_j$
    $\widetilde{\mathbf{x}}_j \Leftarrow \mathbf{x}_i$
    $\widetilde{\mathcal{C}} \Leftarrow \mathcal{H}_C(\widetilde{\mathbf{x}}, \mathbf{W})$
    if $(\widetilde{\mathcal{C}} \succ \mathcal{C}$ **and** $\Phi \succ \mathbf{0})$ **or** $(\widetilde{\mathcal{C}} \prec \mathcal{C}$ **and** $\Phi \prec \mathbf{0})$ **then**
        $\mathbf{x}_i \Leftarrow \widetilde{\mathbf{x}}_i$
        $\mathbf{x}_j \Leftarrow \widetilde{\mathbf{x}}_j$
        $\mathcal{C} \Leftarrow \widetilde{\mathcal{C}}$
    else
        $\Theta \Leftarrow 1$
    end
    $\Omega_t \Leftarrow \mathcal{C}$
end

Algorithm 6: Pseudo code of the greedy homophily rearrangement algorithm for a single categorical generic vertex feature

Chapter 5: Multivariate homophily rearrangement algorithms

As I highlighted previously, it is self-evident in real networks that the vertices have multiple generic features. This also means that homophily might be present regarding multiple features, and the presence of different types of homophily might drive the network evolution and spreading processes on the network in multiple ways. Because of this, generating networks that show prescribed levels of homophily regarding all of the known generic vertex features to obtain reference models. Moreover, it is an important step in order to simulate similarity based diffusion. Importantly, the work of Yavas & Yusel (2014) does not consider the generation of such networks. They only consider univariate systems, while my work is more comprehensive as it considers multivariate homophily rearrangement algorithms. Their simulations of diffusion only consider networks that have a single generic vertex feature. Therefore, it is a novel contribution that one is able to generate networks that show homophily and heterophily regarding multiple features.

The correlation structure of generic vertex features is an important macro-level property of networks. In addition, the correlation of the non-topological node features might show that certain observed homophilous phenomena are possibly endogenous. By switching the values of a single generic vertex feature in order to achieve certain levels of homophily, this correlation structure is prone to be changed. This disorganization of the homophily structure is troublesome if our goal is to understand the homophilous phenomena under the assumption of an unchanged correlation structure. In this section I propose algorithms that keep the network's generic vertex feature correlation structure, the topology, and distributions of generic vertex features unchanged, and rearrange the feature values in a way that target levels of homophily or heterophily regarding multiple variables are achieved. The heuristic multivariate homophily rearrangement algorithm is presented in Subsection 5.1 and the heuristic multivariate homophily rearrangement algorithm with

bag of indices is described in Subsection 5.2. A multivariate implementation of the greedy algorithm is discussed in Section 5.3

5.1 Heuristic algorithm

The operation of the multivariate heuristic homophily rearrangement algorithm is described by pseudo-code in Algorithm 7. The algorithm requires a network described by an adjacency matrix $\mathbf{W}$, and a matrix of generic vertex features here denoted by $\mathbf{X}$. The matrix of generic vertex features has p columns according to the number of generic vertex features, each feature is represented by a column. The target vector of homophily levels are imputed by $(\Phi_1, \ldots, \Phi_p)$, which has scalar an vector elements (according to the type of homophily measurement – categorical or universal). In addition, the algorithm requires an ensemble homophily measurement function which is normalized – for details see Definition 3.15 in Section 3.3 of Chapter 3.

The multivariate heuristic homophily rearrangement algorithm is initialized by counting the vertices in the network – this number equals the number of rows in $\mathbf{X}$. The number of vertices is assigned to be the scalar N. The iterative step counter is set to be zero, an initial homophily profile $(\mathcal{C}_1, \ldots, \mathcal{C}_p)$ is calculated from the initial assignment of feature values by the ensemble homophily measurement function. The time specific homophily profile Ω_t is assigned to be the previously calculated homophily profile. Following the initialization of the rearrangement algorithm an iterative process starts.

The iterative process is controlled by two conditions in the block statement, they are connected by an OR operator. Therefore, only one of the conditions is active. If either of them is satisfied, the iterative process stops. In one case the process stops when the prescribed target homophily vector elements are positive for all of the variables, and the homophily profiles elements are all higher than respective elements of the target homophily vector – when it holds that $(\Phi_1, \ldots, \Phi_p) \preceq (\mathcal{C}_1, \ldots, \mathcal{C}_p)$. In the other case, the imputed target vector described heterophily for each of the variables. The iterative process stops when the elements of the homophily profile are all lower than corresponding elements of the target vector – formally, when $(\Phi_1, \ldots, \Phi_p) \succeq (\mathcal{C}_1, \ldots, \mathcal{C}_p)$. Until the conditions are satisfied, the algorithm iteratively repeats the steps that are recapitulated in the following paragraphs.

1. The iterative step counter is increased by one. The generic feature matrix $\mathbf{X}$ is

assigned to be the temporary feature matrix $\widetilde{\mathbf{X}}$. Later two random integers are chosen from the $[1, N]$ interval, and assigned to be i and j.

2. The following step only takes place if the i^{th} and j^{th} rows of the feature matrix do not equal. Practically, the vertices that were chosen randomly have at least one feature value that is different. The j^{th} row of the feature matrix $\mathbf{X}$ is assigned to be the i^{th} row of the temporary generic feature matrix $\widetilde{\mathbf{X}}$. Similarly, the i^{th} row of the feature matrix $\mathbf{X}$ is assigned to be the j^{th} row of the temporary generic feature matrix. Based on the temporary feature matrix a temporary homophily profile is calculated, this is denoted by $(\widetilde{\mathcal{C}}_1, \ldots, \widetilde{\mathcal{C}}_p)$.

3. These steps only take place if the previous condition was satisfied and in addition one the followings is also true. First, the target vector described homophily, and the elements of the temporary homophily profile are element-wise higher than elements of the homophily profile. Second, the target vector described heterophily, and the elements of the temporary homophily profile are element-wise lower than elements of the homophily profile. If one of these criteria was fulfilled, the i^{th} row of the temporary feature matrix $\widetilde{\mathbf{X}}$ is assigned to be the i^{th} row of the generic feature matrix $\mathbf{X}$. Alike, the j^{th} row of the temporary feature matrix $\widetilde{\mathbf{X}}$ is assigned to be the j^{th} row of the generic feature matrix $\mathbf{X}$. The homophily profile is updated with the temporary homophily profile.

4. The time specific homophily profile Ω_t is set to be equal with the homophily profile irrespective of the previous controlling block statements.

This algorithm does not differ fundamentally from the heuristic algorithms discussed in Section 4.1 of Chapter 4, but a few important observations have to be made about it. The algorithm exchanges all of the generic feature values between the participating vertices. This means that whole rows of the generic vertex feature matrix are exchanged. Exchanges are becoming permanent only if all of the homophily values are increased or decreased. If the generic vertex feature matrix is wide (the number of features is high) the number of effective steps that result in permanent feature value exchanges will be low. Relaxation of the conditions that control permanent feature value exchanges helps this, as simulation results in Section 7.2 of Chapter 7 show. In addition, the homophily levels might diverge from the target value if the target value is achieved in an early iterative step and features are correlated. An R implementation specifically for continuous variables is attached in Appendix E as Script E.9. This specific implementation assumes that the generic vertex features are continuous.

Data: Target homophily profile $(\Phi_1, \ldots, \Phi_p)$, a feature matrix $\mathbf{X}$ with p columns and an adjacency matrix describing the weighted edges in the network denoted by $\mathbf{W}$.

Result: Network with adjacency matrix $\mathbf{W}$, feature matrix $\mathbf{X}$ with p columns where the elements of the homophily profile $(\mathcal{C}_1, \ldots, \mathcal{C}_p)$ in absolute terms are greater than corresponding elements of $(\Phi_1, \ldots, \Phi_p)$.

$N \Leftarrow \left|\mathbf{X}^1\right|$
$t \Leftarrow 0$
$(\mathcal{C}_1, \ldots, \mathcal{C}_p) \Leftarrow \mathcal{E}(\mathcal{H}_1(\mathbf{X}^1, \mathbf{W}), \ldots, \mathcal{H}_p(\mathbf{X}^p, \mathbf{W}))$
$\Omega_t \Leftarrow (\mathcal{C}_1, \ldots, \mathcal{C}_p)$
while $((\Phi_1, \ldots, \Phi_p) \succeq (\mathcal{C}_1, \ldots, \mathcal{C}_p)$ **and** $(\Phi_1, \ldots, \Phi_p) \succ \mathbf{0})$ **or**
    $((\Phi_1, \ldots, \Phi_p) \preceq (\mathcal{C}_1, \ldots, \mathcal{C}_p)$ **and** $(\Phi_1, \ldots, \Phi_p) \prec \mathbf{0})$ **do**
    $t \Leftarrow t + 1$
    $\widetilde{\mathbf{X}} \Leftarrow \mathbf{X}$
    $i \sim U([1, N])$
    $j \sim U([1, N])$
    if $\widetilde{\mathbf{X}}_i \neq \widetilde{\mathbf{X}}_j$ **then**
        $\widetilde{\mathbf{X}}_i \Leftarrow \mathbf{X}_j$
        $\widetilde{\mathbf{X}}_j \Leftarrow \mathbf{X}_i$
        $(\widetilde{\mathcal{C}}_1, \ldots, \widetilde{\mathcal{C}}_p) \Leftarrow \mathcal{E}(\mathcal{H}_1(\widetilde{\mathbf{X}}^1, \mathbf{W}), \ldots, \mathcal{H}_p(\widetilde{\mathbf{X}}^p, \mathbf{W}))$
        if $((\widetilde{\mathcal{C}}_1, \ldots, \widetilde{\mathcal{C}}_p) \succ (\mathcal{C}_1, \ldots, \mathcal{C}_p)$ **and** $(\Phi_1, \ldots, \Phi_p) \succ \mathbf{0})$ **or**
          $((\widetilde{\mathcal{C}}_1, \ldots, \widetilde{\mathcal{C}}_p) \prec (\mathcal{C}_1, \ldots, \mathcal{C}_p)$ **and** $(\Phi_1, \ldots, \Phi_p) \prec \mathbf{0})$ **then**
            $\mathbf{X}_i \Leftarrow \widetilde{\mathbf{X}}_i$
            $\mathbf{X}_j \Leftarrow \widetilde{\mathbf{X}}_j$
            $(\mathcal{C}_1, \ldots, \mathcal{C}_p) \Leftarrow \mathcal{E}(\mathcal{H}_1(\mathbf{X}^1, \mathbf{W}), \ldots, \mathcal{H}_p(\mathbf{X}^p, \mathbf{W}))$
        end
    end
    $\Omega_t \Leftarrow (\mathcal{C}_1, \ldots, \mathcal{C}_p)$
end

Algorithm 7: Pseudo code of the heuristic homophily rearrangement algorithm for a matrix of generic vertex features

5.2 Heuristic algorithm with bag of indices

The bag of indices algorithm is also applicable in multivariate homophily rearrangement problems. They key is again the application of ensemble homophily measurement functions in order to describe global similarities regarding multiple generic vertex features. The algorithm needs a feature matrix $\mathbf{X}$ with p features of interest, a target homophily profile denoted by $(\Phi_1, \ldots, \Phi_p)$ and an adjacency matrix $\mathbf{W}$. The homophily profile elements either have to be all positive or negative – the goal has to be homophily regarding all of the variables or heterophily towards all of them. The imputation of a mixed target profile is not allowed. In addition, an ensemble homophily measurement function $\mathcal{E}(\mathcal{H}_1(\mathbf{X}^1, \mathbf{W}), \ldots, \mathcal{H}_p(\mathbf{X}^p, \mathbf{W}))$ has to be predefined with p homophily measurement functions that are type specific for the corresponding features.

Algorithm 8 summarizes with pseudo-code the heuristic multivariate homophily rearrangement algorithm with bag of indices. The process is initiated by defining the scalar N which equals to the number of elements in the first feature vector. This number is essentially the number of vertices in the system. Based on this number of vertices, $\mathcal{B}$ the bag of indices (a set of integers) is defined which contains all integers from 1 to N. Each of the vertices has a linked index. The time index t has a zero value during the initialization process. The ensemble homophily measurement function is used for calculating an initial homophily profile, which is denoted by $(\mathcal{C}_1, \ldots, \mathcal{C}_p)$. The profile is assigned in place of the time dependent homophily profile Ω_t. The algorithm inchoates an iterative process which is controlled by the homophily target and the actual level of homophily. The iterative operations stop on two occasions. First, if the target profile describes homophily and the homophily level for each variables reaches the corresponding Φ value. Second, if the target is heterophily for all the generic vertex features and the feature specific heterophily levels get at (or get below) the respective Φ values. In the following paragraphs I briefly summarize the steps during the operations that are part of the iterative process.

1. The discrete time ticker t is incremented by one. The temporary feature matrix $\widetilde{\mathbf{X}}$ gets its value from the generic feature matrix $\mathbf{X}$. Two random indices are chosen from the bag of indices – these random indices are assigned in place of i and j.

2. If the temporary features matrices rows $\widetilde{\mathbf{X}}_i$ and $\widetilde{\mathbf{X}}_j$ are not the same the row row exchanges take place between the temporary feature matrix and the feature matrix. The j^{th} row of the feature matrix replaces the i^{th} row of the temporary feature matrix. Similarly, the i^{th} row of the feature matrix replaces the j^{th} row of

the temporary feature matrix. After the feature row swap, $\widetilde{\mathbf{X}}$ is used in order to obtain a temporary homophily profile. This homophily profile is used for evaluating whether the exchange was effective.

3. The exchange of feature matrix rows is effective and permanent on two occasions. First, if the temporary homophily profile, which is denoted by $(\widetilde{\mathcal{C}}_1, \ldots, \widetilde{\mathcal{C}}_p)$, is element-wise greater than $(\mathcal{C}_1, \ldots, \mathcal{C}_p)$ and the target is homophily for each of the features. Second, if the target is heterophily for each of the features and the $(\widetilde{\mathcal{C}}_1, \ldots, \widetilde{\mathcal{C}}_p)$ is element-wise smaller than $(\mathcal{C}_1, \ldots, \mathcal{C}_p)$. When one of these conditions is satisfied, the i^{th} row of the feature matrix becomes the i^{th} row of the temporary feature matrix. Likewise, the i^{th} row of the feature matrix replaces the i^{th} row of the temporary feature matrix. The new feature matrix is used for calculating the homophily profile. As a result of the effective feature matrix row exchange, the indices i and j are removed from the bag of indices.

4. Independent from the conditions on feature matrix row exchanges, the time specific homophily profile is defined to be $(\mathcal{C}_1, \ldots, \mathcal{C}_p)$ in each iterative step.

Again, the heuristic homophily rearrangement algorithms that use a bag of indices are prone to be stalled because of the bag reduction step – each vertex can participate in one permanent exchange of features. The severeness of this problem increases with relaxation of the exchange conditions, feature exchanges are allowed that do not move the homophily levels closer to the target values. However, the algorithm can be augmented with a time dependent control exit flag – if the time index is above a certain number, the exit flag returns with a value that stops the iterative process. In this way the algorithm might exit with a homophily profile that is not reaching the target value, but the algorithm does not end up an infinite loop. This augmentation is a possible extension of all the heuristic algorithms (both in case of univariate and multivariate ones). The algorithm can be reinitialized with a new bag of indices after such exit. In this case it would not be fundamentally different from the simple heuristic homophily rearrangement algorithm.

The R implementation of the multivariate heuristic homophily rearrangement algorithm with bag of indices is enclosed as Script E.10 in Appendix E. Similarly to the the baseline heuristic multivariate homophily rearrangement algorithm the features are assumed to be Gaussian.

Data: Target homophily profile $(\Phi_1, \ldots, \Phi_p)$, a feature matrix $\mathbf{X}$ with p columns and an adjacency matrix describing the weighted edges in the network denoted by $\mathbf{W}$.

Result: Network with adjacency matrix $\mathbf{W}$, feature matrix $\mathbf{X}$ with p columns where the elements of the homophily profile $(\mathcal{C}_1, \ldots, \mathcal{C}_p)$ in absolute terms are greater in absolute terms than corresponding elements of $(\Phi_1, \ldots, \Phi_p)$.

$N \Leftarrow \left|\mathbf{X}^1\right|$
$\mathcal{B} \Leftarrow \{1, \ldots, N\}$
$t \Leftarrow 0$
$(\mathcal{C}_1, \ldots, \mathcal{C}_p) \Leftarrow \mathcal{E}(\mathcal{H}_1(\mathbf{X}^1, \mathbf{W}), \ldots, \mathcal{H}_p(\mathbf{X}^p, \mathbf{W}))$
$\Omega_t \Leftarrow (\mathcal{C}_1, \ldots, \mathcal{C}_p)$
while $((\Phi_1, \ldots, \Phi_p) \succeq (\mathcal{C}_1, \ldots, \mathcal{C}_p)$ **and** $(\Phi_1, \ldots, \Phi_p) \succ \mathbf{0})$ **or**
    $((\Phi_1, \ldots, \Phi_p) \preceq (\mathcal{C}_1, \ldots, \mathcal{C}_p)$ **and** $(\Phi_1, \ldots, \Phi_p) \prec \mathbf{0})$ **do**
    $t \Leftarrow t + 1$
    $\widetilde{\mathbf{X}} \Leftarrow \mathbf{X}$
    $i \sim U(\mathcal{B})$
    $j \sim U(\mathcal{B})$
    if $\widetilde{\mathbf{X}}_i \neq \widetilde{\mathbf{X}}_j$ **then**
        $\widetilde{\mathbf{X}}_i \Leftarrow \mathbf{X}_j$
        $\widetilde{\mathbf{X}}_j \Leftarrow \mathbf{X}_i$
        $(\widetilde{\mathcal{C}}_1, \ldots, \widetilde{\mathcal{C}}_p) \Leftarrow \mathcal{E}(\mathcal{H}_1(\widetilde{\mathbf{X}}^1, \mathbf{W}), \ldots, \mathcal{H}_p(\widetilde{\mathbf{X}}^p, \mathbf{W}))$
        if $((\widetilde{\mathcal{C}}_1, \ldots, \widetilde{\mathcal{C}}_p) \succ (\mathcal{C}_1, \ldots, \mathcal{C}_p)$ **and** $(\Phi_1, \ldots, \Phi_p) \succ \mathbf{0})$ **or**
            $((\widetilde{\mathcal{C}}_1, \ldots, \widetilde{\mathcal{C}}_p) \prec (\mathcal{C}_1, \ldots, \mathcal{C}_p)$ **and** $(\Phi_1, \ldots, \Phi_p) \prec \mathbf{0})$ **then**
            $\mathbf{X}_i \Leftarrow \widetilde{\mathbf{X}}_i$
            $\mathbf{X}_j \Leftarrow \widetilde{\mathbf{X}}_j$
            $(\mathcal{C}_1, \ldots, \mathcal{C}_p) \Leftarrow \mathcal{E}(\mathcal{H}_1(\mathbf{X}^1, \mathbf{W}), \ldots, \mathcal{H}_p(\mathbf{X}^p, \mathbf{W}))$
            $\mathcal{B} \Leftarrow \mathcal{B} \setminus \{i, j\}$
        end
    end
    $\Omega_t \Leftarrow (\mathcal{C}_1, \ldots, \mathcal{C}_p)$
end

Algorithm 8: Pseudo code of the heuristic homophily rearrangement algorithm with bag of indices for a matrix of generic vertex features

5.3 Greedy algorithm

The extension of the homophily rearrangement algorithms to multivariate systems is the most complex in case of the greedy algorithm. The vertex index selection heuristic directly takes advantage of the feature's type (categorical or not). However, with inducing a random element, the proper multivariate extension of this homophily rearrangement algorithm also becomes possible. The greedy multivariate homophily rearrangement algorithm is described with pseudo-code by Algorithm 9.

Similarly to the other multivariate homophily rearrangement algorithms, this one needs a generic vertex feature matrix $\mathbf{X}$ with p features, which might contain multiple types of variables. The algorithm also needs a $\mathbf{W}$ matrix, which describes the relationships among vertices, and a target homophily profile $(\Phi_1, \ldots, \Phi_p)$. Again, this target homophily profile either contains only positive or negative elements, but not a mixture of positive and negative elements. In order to measure the macro-level similarities properly the algorithm uses an ensemble homophily measurement function which has to be normalized. All of the ensemble homophily measurement functions have to be type specific – universal or categorical, according to the type of the feature. Furthermore, the algorithm needs to define a specific vertex selection heuristic. One of the selection heuristics should be able to treat a single non-categorical feature ($\mathcal{S}_U$), while the other heuristic should be able to deal with a single categorical generic vertex feature ($\mathcal{S}_C$). In this case, auxiliary vertex selection heuristics are chosen to be, respectively, Algorithms 11 and 12 in Appendix B.

The greedy multivariate homophily rearrangement algorithm is initiated by setting the exit parameter Θ to be zero. The discrete time ticker starts with a zero value. The set $\mathcal{P}$ contains the indices of the generic vertex features – essentially the integers from 1 to p, which is the number of features. Based on the initial set up features with the ensemble homophily function, a homophily profile is calculated. The homophily profile $(\mathcal{C}_1, \ldots, \mathcal{C}_p)$ is assigned in place of the time specific homophily profile Ω_t. Following these early steps, the algorithm starts an iterative procedure in order to achieve the preset homophily levels. This procedure stops when the values in the homophily profile are higher or are at the target homophily profile values. It also terminates when the heterophily values in the heterophily profile are below or at the heterophily values in the heterophily target profile. It also ends when the exit flag takes a value different from one. The next paragraphs give a sketch of the above mentioned iterative steps.

1. The time ticker is incremented by one, the feature matrix $\mathbf{X}$ is set to be the temporary feature matrix $\widetilde{\mathbf{X}}$. A random integer k is chosen from the set P. The integer k is used for choosing a generic vertex feature from the feature matrix.

2. Based on the type of the randomly chosen generic vertex feature, there are two possibilities. If the feature is categorical or ordinal, the categorical vertex index selection algorithm is applied to choose a pair of vertices for feature value exchange in a greedy way. In the other case, the universal vertex index selection is applied in order to choose a pair of indices. It is substantial that the choice of indices is only affected by $\mathbf{X}^k$ and the corresponding homophily/heterophily target. Later the feature value exchange happens for all of the features between the pair of nodes.

3. The chosen pair of vertex indices is used for exchanging the rows of the feature matrix. The i^{th} row of the temporary feature matrix is overwritten by the j^{th} row of the feature matrix. Just like this, the j^{th} row of the temporary feature matrix is replaced by the i^{th} row of the feature matrix. The temporary feature matrix $\widetilde{\mathbf{X}}$ is used for calculating a temporary homophily profile.

4. Exchange of feature matrix rows is made to be permanent by two main steps: by replacing the feature matrices i^{th} and j^{th} row with the i^{th} and j^{th} row of the temporary feature matrix. This happens on two occasions. First, if the target of the rearrangement is homophily for each of the variables and the homophily levels are increased. Second, if the the target of the rearrangement process is heterophily and the homophily values are all decreased. Importantly, if the conditions are satisfied, the temporary homophily profile replaces the actual homophily profile. Otherwise, the exit flag receives a value equal to one and the algorithm stops.

5. Independent from the above listed criteria and conditions the time specific homophily profile receives its value from the homophily profile.

The choice of indices is only based on potential homophily or heterophily based on the randomized feature. However, the goal of the algorithm is to achieve homophily regarding a number of variables not just the one that is randomized in the vertex selection process. Because of this, individual steps are only greedy towards the feature that is chosen in the iterative step. The greediness of the algorithm is not ensured with respect to other generic vertex features in a specific step. However, the random nature of the feature selection ensures that the probability of being greedy towards a certain feature is the same for all of the features in each iterative step.

Data: Target homophily profile $(\Phi_1, \ldots, \Phi_p)$, a feature matrix $\mathbf{X}$ with p columns and matrix $\mathbf{W}$ describing the weighted edges in the network.

Result: Network with adjacency matrix $\mathbf{W}$, feature matrix $\mathbf{X}$ with p columns where the elements of the homophily profile $(\mathcal{C}_1, \ldots, \mathcal{C}_p)$ in absolute terms are greater than corresponding elements of $(\Phi_1, \ldots, \Phi_p)$.

$\Theta \Leftarrow 0$
$t \Leftarrow 0$
$P \Leftarrow \{1, \ldots, p\}$
$(\mathcal{C}_1, \ldots, \mathcal{C}_p) \Leftarrow \mathcal{E}(\mathcal{H}_1(\mathbf{X}^1, \mathbf{W}), \ldots, \mathcal{H}_p(\mathbf{X}^p, \mathbf{W}))$
$\Omega_t \Leftarrow (\mathcal{C}_1, \ldots, \mathcal{C}_p)$
while $[((\Phi_1, \ldots, \Phi_p) \succeq (\mathcal{C}_1, \ldots, \mathcal{C}_p)$ **and** $(\Phi_1, \ldots, \Phi_p) \succ \mathbf{0})$ **and** $(\Theta = 0)]$ **or**
    $[((\Phi_1, \ldots, \Phi_p) \preceq (\mathcal{C}_1, \ldots, \mathcal{C}_p)$ **and** $(\Phi_1, \ldots, \Phi_p) \prec \mathbf{0})$ **and** $(\Theta = 0)]$ **do**
    $t \Leftarrow t + 1$
    $\widetilde{\mathbf{X}} \Leftarrow \mathbf{X}$
    $k \Leftarrow U(P)$
    if $\mathbf{X}^k$ **is ordinal or categorical then**
        $\{i, j\} \Leftarrow \mathcal{S}_C(\Phi_k, \mathbf{X}^k, \mathbf{W})$
    else
        $\{i, j\} \Leftarrow \mathcal{S}_U(\Phi_k, \mathbf{X}^k, \mathbf{W})$
    end
    $\widetilde{\mathbf{X}}_i \Leftarrow \mathbf{X}_j$
    $\widetilde{\mathbf{X}}_j \Leftarrow \mathbf{X}_i$
    $(\widetilde{\mathcal{C}}_1, \ldots, \widetilde{\mathcal{C}}_p) \Leftarrow \mathcal{E}(\mathcal{H}_1(\widetilde{\mathbf{X}}^1, \mathbf{W}), \ldots, \mathcal{H}_p(\widetilde{\mathbf{X}}^p, \mathbf{W}))$
    if $((\widetilde{\mathcal{C}}_1, \ldots, \widetilde{\mathcal{C}}_p) \succ (\mathcal{C}_1, \ldots, \mathcal{C}_p)$ **and** $(\Phi_1, \ldots, \Phi_p) \succ \mathbf{0})$ **or**
      $((\widetilde{\mathcal{C}}_1, \ldots, \widetilde{\mathcal{C}}_p) \prec (\mathcal{C}_1, \ldots, \mathcal{C}_p)$ **and** $(\Phi_1, \ldots, \Phi_p) \prec \mathbf{0})$ **then**
        $\mathbf{X}_i \Leftarrow \widetilde{\mathbf{X}}_i$
        $\mathbf{X}_j \Leftarrow \widetilde{\mathbf{X}}_j$
        $(\mathcal{C}_1, \ldots, \mathcal{C}_p) \Leftarrow \mathcal{E}(\mathcal{H}_1(\mathbf{X}^1, \mathbf{W}), \ldots, \mathcal{H}_p(\mathbf{X}^p, \mathbf{W}))$
    else
        $\Theta \Leftarrow 0$
    end
    $\Omega_t \Leftarrow (\mathcal{C}_1, \ldots, \mathcal{C}_p)$
end

Algorithm 9: Pseudo code of the heuristic homophily rearrangement algorithm with bag of indices for a matrix of generic vertex features

Due to the multivariate nature of the problem and the stringent permanent feature value switching conditions, the algorithm returns an exit flag equal to one numerous times. Nevertheless, due to the random nature of variable and selection the homophily rearrangement algorithm is restartable. If the algorithm stops with an exit flag different from zero, it can be restarted until it is stalled totally (none of the features can move the homophily levels towards the target values). It has to be underlined that multivariate homophily rearrangement algorithms are combinable, this connotes that when one of the algorithms is stalled an other algorithm can take the partial solution as an input. With a series of *algorithm pipelines* that pass to each other partial solutions, one might solve complex – high dimensional homophily rearrangement problems.

Chapter 6: Similarity based diffusion

The majority of models which investigate the process of diffusion on networks does not consider that the diffusion might depend on similarity of the nodes participating in the spreading. Those models that do consider similarity based diffusion, such as Halberstam & Knight (2014); Yavas & Yusel (2014), are limited in multiple ways as discussed in Section 2.3 of Chapter 1. The novelty of the similarity based diffusion model introduced in this chapter lies in three important features.

First, in my model the similarity of vertices is measured by distances which can describe dissimilarity regarding multiple features that have different types (binary, ordinal, categorical or continuous). Because of this, when vertices are multi-dimensional, similarity regarding multiple generic vertex features can be taken into account in order to quantify the likeliness of spreading. Moreover, this allows for investigating how correlated generic vertex features influence spreading on the network. Second, the spreading mechanism in earlier models is different from the mechanism applied in my model. Earlier models used either the threshold model (Granovetter, 1978) with a relative adaptation threshold (Watts, 2002) or modifications of the model proposed by Bass (1969) to capture similarity based adaptation dynamics on the network. Third, with the use of the introduced homophily rearrangement algorithms, the similarity based diffusion process can be simulated on networks that have the same topology, homophily regarding the features, correlation structure and distribution of these features.

The remainder of the chapter is structured as follows. In Section 6.1 I discuss the pairwise transmission probability equations that are a core idea of the similarity based diffusion model introduced. The idea of heterogeneous pairwise transmission probability weighting is presented in Section 6.2, and the chapter concludes with the similarity based susceptible-infected model itself in Section 6.3.

6.1 Pairwise transmission probability equations

In the baseline setup we consider a network which is essentially a non directed graph, with multiple generic vertex features (considering a single vertex feature would not make a difference). This network is described by a weighted adjacency matrix $\mathbf{W}$ and a generic feature matrix $\mathbf{X}$. This matrix contains features that have different types – these features are possibly binary, categorical, ordinal, count or continuous. The vertices in the network represent agents and the edges among the vertices represent the links between the agents. It is assumed throughout the thesis that the topology is fixed. In this setting, it simply means that agents cannot form new ties, and agents do not appear or disappear in the system. In addition, the model has a simple discrete time dimension, where the time periods are indexed by t.

In addition to their generic features the agents can obtain an information from outside the system and also from other agents. The agents who do not receive this information from outside the network are the seeders. An agent who received the information from outside in the first time period is an initial seeder. This above mentioned information can take the form of a gossip, first contact with a technological innovation or intelligence about labor market opportunities. Knowledge about the information in time period t is described by the binary vector $\mathbf{Y}$. The number of elements in the vector equals to the number of vertices in the network. State of agent i regarding the information in time period t can be described as:

$$\mathbf{Y}_{t,i} = \begin{cases} 1, \text{ if agent } i \text{ received the information in period } t. \\ 0, \text{ otherwise} \end{cases}$$

Let us imagine that we have two agents, respectively denoted with the indices i and j. These agents can interact with each other, because there is a link between them. At time period t node i already received the information, while node j did not. In terms of the susceptible-infected model node i is infected, while node j is susceptible. The probability that in a certain time period node i passes to j the information is denoted by $P_{i,j}$. This probability can be expressed as a function of dissimilarity between the two nodes – for a single feature this is described by Equation (6.1).

$$P_{i,j} = P_0 \cdot \underbrace{\Psi(-\gamma \cdot d(\mathbf{X}_i, \mathbf{X}_j))}_{\text{Base function}} \tag{6.1}$$

In the remainder of the paper I reference Equation (6.1) as the pairwise transmission probability equation. In a pairwise transmission probability equation P_0 denotes the baseline

transmission probability of the information or infection. The base function Ψ depends on two factors:

1. First, it depends on the sensitivity coefficient γ, which is a global control parameter – for each considered pairwise relationships between two agents it is assumed to be the same.[1] The γ parameter is strictly positive.

2. Second, it depends on the dissimilarity between agent i's and j's features, which is described by $d(\mathbf{X}_i, \mathbf{X}_j)$ – this is a simple canonical dissimilarity metric (Deza & Deza, 2009). For example, if all of the features are continuous the dissimilarity can be described by the Euclidean distance of two agent's features. But at the same time if the feature matrix contains categorical or ordinal variables the dissimilarity metric has to be chosen in a way that it can describe dissimilarity regarding such features. The metric proposed by Gower (1971) is able to include ordinal features in the dissimilarity measurement.

Importantly, the first order derivative of the base function regarding the dissimilarity of agents satisfies Inequality 6.2.

$$\frac{\partial \Psi(-\gamma \cdot d(\mathbf{X}_i, \mathbf{X}_j))}{\partial d(\mathbf{X}_i, \mathbf{X}_j)} < 0 \tag{6.2}$$

Practically Inequality (6.2) means that the value of the base function decreases as the dissimilarity between two agents increases. Conversely, if the dissimilarity decreases between two agent's, the value of the base function increases. Moreover, the base function Ψ has to be be characterized by the two important limits. These are described by Equations (6.3) and (6.4).

$$\lim_{d(\mathbf{X}_i, \mathbf{X}_j) \to \infty} \Psi(-\gamma \cdot d(\mathbf{X}_i, \mathbf{X}_j)) = 0 \tag{6.3}$$

$$\lim_{d(\mathbf{X}_i, \mathbf{X}_j) \to 0} \Psi(-\gamma \cdot d(\mathbf{X}_i, \mathbf{X}_j)) = 1 \tag{6.4}$$

Inequality 6.2 combined with Equations (6.3) and (6.4) ensure that the base function has a $[0, 1]$ range. A potential base function which satisfies the above mentioned criteria is the exponential. The sensitivity analysis of the pairwise transmission probability values can be summarized as follows:

1. The transmission probability increases if the baseline transmission probability is higher. Reversely, if the baseline transmission probability is lower, the pairwise transmission probability will also be lower.

[1] It is essential to point out that this assumption is relaxed in the next section.

2. If the sensitivity to dissimilarity increases the pairwise transmission probability decreases (this affects the whole system). Likewise, if the sensitivity to dissimilarity decreases, the transmission probabilities increase.

3. Finally, the higher the dissimilarity is between two agents the lower the information transmission probability will be between them. The higher similarity between vertices results in higher transmission probabilities.

Remark 6.1. *Because the pairwise dissimilarity between agents i and j satisfies the non-negativity axiom it is always true that the baseline transmission probability is not smaller than any pairwise transmission probability in the system – basically this means that $P_0 \geq P_{i,j}$.*

The base function's range is constrained to the $[0, 1]$ interval. From this it comes that $P_{i,j}$ is the highest when the base functions value is one, otherwise, it is smaller than 1. In the first case, the pairwise transmission probability, which is the product of P_0 and base function value, will be smaller than P_0.

Remark 6.2. *If the pairwise transmission between agents i and j equals to the baseline transmission probability, then agents i and j have the same feature values. The reversal is also true. If two agents share the same feature values, then the pairwise transmission probability between them is the baseline transmission probability.*

The phenomenon that Remark 6.2 discusses results from the identity of indiscernibles. If agents i and j have the same features, we know that $\mathbf{X}_i = \mathbf{X}_j$. In this case the distance of features satisfies that $d(\mathbf{X}_i, \mathbf{X}_j) = 0$. Based on Equation (6.4), we know that the value of the base function equals to one when the distance between two agents' features is zero. The baseline transmission probability P_0 multiplied by one is the baseline transmission probability itself. Proving that the reversal is true is straightforward.

$$P_0 = P_{i,j} \Leftrightarrow \mathbf{X}_i = \mathbf{X}_j$$

Remark 6.3. *The pairwise transmission probability of agent i transmitting the information to j if only i has the information equals to the probability that j passes the information to i when only j has the information. Simply it is true that the pairwise transmission probabilities between two agents satisfy that $P_{i,j} = P_{j,i}$.*

The property described by Remark 6.3 comes from the the symmetry property of the feature distances. We know that the feature distances satisfy that $d(\mathbf{X}_i, \mathbf{X}_j) = d(\mathbf{X}_j, \mathbf{X}_i)$. The base function is universal for the agents and also in this model it is assumed that

the sensitivity of agents to the feature distances is the same. From this it also comes that the pairwise transmission probabilities are equal to each other.

6.2 Asymmetric weighting of dissimilarity

The basic pairwise transmission probability approach which is the subject of Section 6.1 assumes that agents share a universal sensitivity to dissimilarities. This means that $\gamma_i = \gamma_j$ for any pair of agents i and j. However, this is a considerably strong assumption about the behavior of agents. It is a well-founded assumption that the same social tie is asymmetrically up-weighted and down-weighted by agents who differ in a certain (categorical, ordinal or binary) trait. For example, the importance of a social link (in terms of gaining information) in a network of friendships might be different for two students who only differ in their race.

Let us imagine the we have a network which is characterized by either an adjacency matrix $\mathbf{W}$ or by the set of edges and vertices such that $\mathrm{G}(V, E)$. Furthermore, let us assume that vertices in the network represent agents and links between them to allow spreading information. The agents have generic features and these features are represented in the matrix $\mathbf{X}$, where columns denote different features and the indexed rows describe the specific agent's features. The feature $\mathbf{x}$ is a column of matrix $\mathbf{X}$. In addition it is binary, ordinal or categorical and it has $1, \ldots, m$ unique values. The sensitivity of agent i to the dissimilarity of other agents is a function of its respective $\mathbf{x}$ value. The different breeds of agents (regarding feature $\mathbf{x}$) have a shared value of sensitivity, so agents who have a given level of $\mathbf{x}$ have the same sensitivity to dissimilarity. Formally this is expressed by Equation 6.5.

$$\gamma_i = \begin{cases} \Gamma_1, \text{ if } \mathbf{x}_i = 1 \\ \quad \vdots \\ \Gamma_m, \text{ if } \mathbf{x}_i = m \end{cases} \tag{6.5}$$

Parameter values $\Gamma_1, \ldots, \Gamma_m$ denote the group specific sensitivities to dissimilarity. This idea, namely that the heterogeneous weighting of dissimilarity is a function of generic vertex features, can be generalized to continuous generic vertex features and to multivariate system (the sensitivity is a function of multiple generic vertex features). Nevertheless, to make my thesis self contained I only included the case when the sensitivity to dissimilarity is a function of a single non continuous generic vertex feature. The properties of pairwise

transmission probabilities which were stated in Remarks 6.1 and 6.2 hold even when the sensitivity to dissimilarity is not the same for every agent. At the same time Remark 6.3 will not hold anymore.

Remark 6.4. *The pairwise transmission probabilities between agents i and j does not necessarily satisfy that $P_{i,j} = P_{j,i}$ if the sensitivity to dissimilarity is weighted in an asymmetric way.*

Let us consider that agents i and j have the same generic vertex feature values except for $\mathbf{x}$. Furthermore, let us assume that the sensitivity to dissimilarity is a function of $\mathbf{x}$, which is a non continuous generic vertex feature. The non equivalence of the feature values simply means that $x_i \neq x_j$, which implies that $\Gamma_i \neq \Gamma_j$, from this it follow that the pairwise transmission probabilities $P_{i,j}$ and $P_{j,i}$ will not be the same. A straightforward consequence of this phenomenon is that if one of the groups has a high sensitivity to dissimilarity and the initial seeder belongs to this sensitive group, the information might get stuck in the seeders group. This has troublesome implications about the spread of innovations or labor market informations, intrinsically discrimination against the other group(s) slows down the spreading of information.

6.3 The similarity based susceptible-infected model

This section introduces the similarity based diffusion model, it is a simple modification of the susceptible-infected diffusion model discussed in (Easley & Kleinberg, 2010; Jackson, 2010; Barabási, 2016). My description of the model and algorithm is general enough to allow for asymmetric weighting of dissimilarity. The similarity based susceptible-infected diffusion model is summarized with pseudo-code by Algorithm 10. The model works on a network which has a predefined topology (this is either defined by the sets of edges and vertices or the adjacency matrix) and a number of generic vertex features. The delineated algorithm assumes, in addition, that the baseline transmission probability, the base function Ψ, the sensitivity parameter(s) and the distance metric are predefined. Sensitivity to the dissimilarity is either universal for the vertices or vertex specific to describe heterogeneous transmission probabilities among nodes. This is why the sensitivity parameter has an index – each of the respective agents has an own sensitivity to the dissimilarity. The initial number of seeders is one, only a single node is infected or has the information – this aspect of the model is easily modifiable, but I have to emphasize it.

The model is initiated by calculating the number of vertices in the system. The number of vertices equals to the length of the first feature vector, this is the scalar N. The convergence to a fully infected system (where every one has the information) is an interesting aspect, his is why I measure the time. Initially, the discrete time index t is set to be zero. Later it will be incremented in each iteration. A random vertex index k is drawn from the set of vertex indices, the corresponding vertex is the initial seeder in the system. This random choice of the initial seeder is substitutable with targeted choice of a vertex. Because initially all of the vertices are in a not infected state, their initial state of having the information is formalized by setting $\mathbf{Y}_0$ to be a vector of zeros. Based on the chosen vertex the k^{th} component of $\mathbf{Y}_0$ is set to be one. Nodes that have the information are in the set $\widetilde{\mathcal{B}}$, while the nodes that do not have the information are in set $\mathcal{B}$.

The spreading process stops when all of the nodes received the information – formally this is satisfied when the number of elements in $\widetilde{\mathcal{B}}$ equals to the number of vertices in the system. Otherwise, if the above mentioned condition is not satisfied, an iterative process starts. First, the set of newly infected nodes, which is noted by $\mathcal{K}$, is set to be an empty set and the time index is incremented by one. In case of each vertex that is infected, the infection is transferable to other vertices that are neighboring the infected ones and they are not infected at time period t. This is done by a double for loop in the implementation, and the exact process is summarized in the following.

The first for loop iterates through the infected nodes – the infected nodes are indexed by i. For each of the infected vertices an empty initial set of infected neighboring nodes is created, this is denoted by the set $\mathcal{I}$. In addition, the set of i's neighbors is obtained with the neighborhood function, this index specific set is $\widetilde{\mathcal{N}}_i$. Later one can iterate through the indices of i's neighboring vertices with ease, the second for loop does exactly this. For each of the neighboring indices a value is drawn randomly from a continuous uniform distribution in the zero-one interval. Based on the pairwise transmission probability equation between nodes i and j, a pair specific transmission probability is calculable. If the transmission probability is higher than the random number drawn from the uniform distribution and the vertex is not already in the set of infected vertices, the neighbor is added to the set of possibly infected neighbors. In addition, its time specific infection state is set to be one – in the pseudo code this is shown by setting $Y_{t,j}$ equal to one.

Data: Network defined by an adjacency matrix or by sets of edges and vertices with generic vertex features in matrix $\mathbf{X}$.

Result: Time dependent infection states for each of the vertices stored in $\mathbf{Y}_t$. The characteristic convergence time to a fully infested state denoted by t.

$N \Leftarrow \left|\mathbf{X}^1\right|$
$t \Leftarrow 0$
$k \Leftarrow$ random index from $1, \ldots, N$
$\mathbf{Y}_0 \Leftarrow \mathbf{0}$
$\mathbf{Y}_{0,k} \Leftarrow 1$
$\mathcal{Y}_t \Leftarrow 1$
$\widetilde{\mathcal{B}} \Leftarrow \{k\}$
$\mathcal{B} \Leftarrow \{1, \ldots, N\} \setminus \widetilde{\mathcal{B}}$
while $N \neq \mathcal{Y}_t$ **do**
    $\mathcal{K} \Leftarrow \emptyset$
    $t \Leftarrow t + 1$
    for i **in** $\widetilde{\mathcal{B}}$ **do**
        $\mathcal{I} \Leftarrow \emptyset$
        $\widetilde{\mathcal{N}}_i \Leftarrow \mathcal{N}_{\mathcal{G}}(i)$
        for j **in** $\widetilde{\mathcal{N}}_i$ **do**
            $P \sim U([0, 1])$
            $P_{i,j} \Leftarrow P_0 \cdot \Psi(-\gamma_i \cdot d(\mathbf{X}_i, \mathbf{X}_j))$
            if $j \notin \widetilde{\mathcal{B}}$ **and** $P_{i,j} > P$ **then**
                $\mathcal{I} \Leftarrow \mathcal{I} \cup \{j\}$
                $\mathbf{Y}_{t,j} \Leftarrow 1$
            end
        end
        $\mathcal{K} \Leftarrow \mathcal{K} \cup \mathcal{I}$
    end
    $\mathcal{B} \Leftarrow \mathcal{B} \setminus \mathcal{K}$
    $\widetilde{\mathcal{B}} \Leftarrow \widetilde{\mathcal{B}} \cup \mathcal{K}$
    $\mathcal{Y}_t \Leftarrow \left|\widetilde{\mathcal{B}}\right|$
    $\mathbf{Y}_{t+1} \Leftarrow \mathbf{Y}_t$
end

Algorithm 10: The similarity based diffusion model

This algorithm design allows for time dependent tracking of the node specific infection during the diffusion process. Importantly in each iteration possibly a new node is added to $\mathcal{I}$ and at the end of the iterative process it contains the nodes infected by vertex i. Before moving to the next infected node the set of nodes that is infected by i is added to the set of newly infected nodes – this is expressed by the union of the index sets $\mathcal{K}$ and $\mathcal{I}$. When the double *for loop* terminates the set of nodes that was infected in time period t is obtained. From the next time period these nodes will be in the set of seeders, who can pass the information to other nodes, so the newly infected nodes are added to the set $\widetilde{\mathcal{B}}$. Moreover, they will not be part of the set of non-infected nodes, because of this they are removed from $\mathcal{B}$. With the new set of seeders one can calculate the time dependent number of infected nodes. The iterations are stopped when all of the nodes have the information (infected). Convergence to the previously mentioned state, where all of the nodes are infected, is only achieved under the assumption that the baseline probability of spreading is non zero, formally, if $P_0 > 0$ and the network has only a single component.

The algorithm can be enriched in a way that time dependent properties of the network can be measured. In each time period one can measure vertex, edge and system specific properties of the networks described by the sets $\widetilde{\mathcal{B}}$ and $\mathcal{B}$. The iterative generation of time specific infected vertex sets $\mathcal{K}$ while time and index specific sets, such as $\mathcal{I}$, allow for observing the role and importance of nodes during the spreading process. The simulation results of similarity based diffusion are discussed in Chapter 8 while the algorithm's R implementation is enclosed in Appendix E as Script E.11.

Chapter 7: Feature rearrangement simulations

This chapter presents the simulation results that were obtained by applying the homophily rearrangement algorithms. The findings support that the algorithms introduced in my thesis are able to tackle the task of homophily rearrangement for both single feature and multivariate networks. Results support that the algorithms are sensitive to certain parameters, such as the target homophily level, the size of the system and the initial distribution of the generic vertex feature. First, I analyze the simulation results generated with univariate homophily rearrangement simulations in Section 7.1. Second, the results of homophily rearrangement simulation on multivariate networks are presented and discussed in Section 7.2.

7.1 Univariate simulations

There are three main goals of the univariate homophily rearrangement simulations. First, I establish with my results that the proposed algorithms are able to generate networks that exhibit preset levels of homophily or heterophily – this is supported by the evidence in Subsections 7.1.1 and 7.1.2. Second, as I mentioned previously the expected convergence time is dependent on multiple properties of the network, namely, the generic vertex feature vector and the target homophily level. In order to show these dependencies I implemented simulations that show the sensitivity of the expected convergence time. Results obtained from these simulations are analyzed in Subsection 7.1.1. Third, it is evident that the proposed heuristic homophily rearrangement algorithms have a random element, this results in that the rearranged generic feature vectors are uncorrelated with each other. Essentially, simulation results regarding this phenomenon and its consequences are investigated in Subsection 7.1.2.

7.1.1 General sensitivity analysis

The fact that the homophily rearrangement algorithms are functional can be proved by a single simulation run on an arbitrarily chosen network. The subfigures of Figure 7.1 were obtained by the application of the heuristic homophily rearrangement algorithm. The topology of the network is described by a Barabási-Albert graph with 1000 nodes, 2 arriving connections and the power parameter is unit (Barabási & Albert, 2002). The vertices only have a single generic vertex feature, which is denoted by $\mathbf{x}$. The actual generic vertex feature was generated by a binary random variable which had a Bernoulli distribution with a parameter equal to 0.5. The R script that was used for the simulation and plotting is attached as Script E.25 in Appendix E. The algorithm used for the rearrangement was the heuristic homophily rearrangement algorithm and Moran's I was used as the homophily measurement function. The target homophily level was prescribed to be 0.6. In Subfigures 7.1a and 7.1b, the binary feature values are represented by the blue and red feature values. The emerging homophily is clearly visible, the vertices on Subfigure 7.1b are likely to have connections with other vertices that have the same color (and feature value). A visualization of another experiment that supports the effectiveness of the algorithm is attached in Appendix D as Figure D.1. In that experiment the topology is described by a fully connected Erdős-Rényi graph (Erdős & Rényi, 1960) with 1000 vertices and 3000 edges. The generic vertex feature that is rearranged had a Bernoulli distribution with $p = 0.5$.

(a) Unordered state

(b) Homophilous state

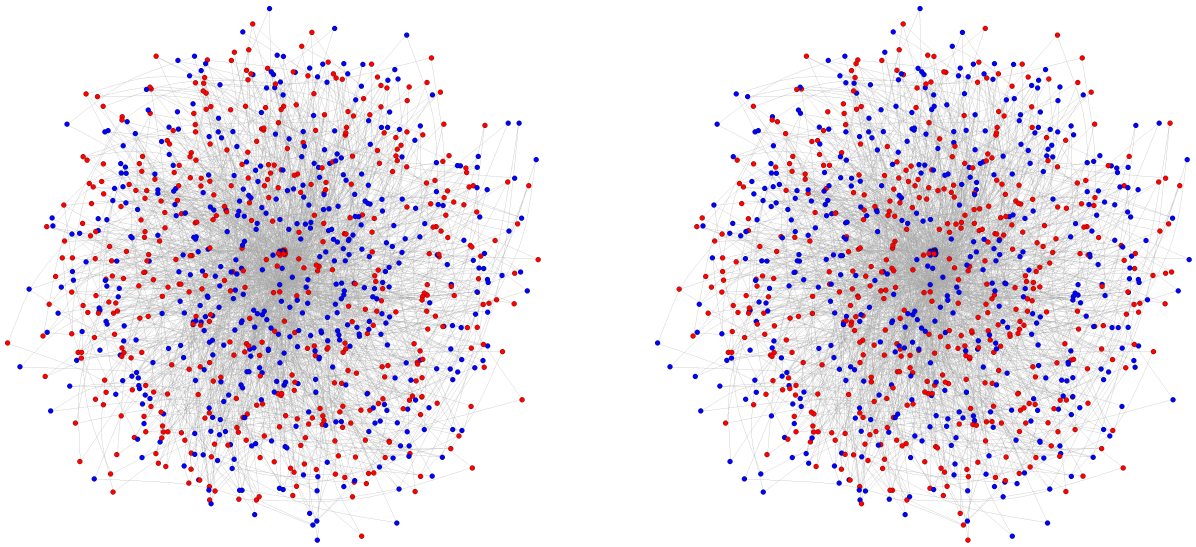


Figure 7.1: Heuristic homophily rearrangement of a binary feature

System size and feature distribution

The previous simulations on the networks which have a a topology described by the Barabási-Albert Model and the Erdős-Rényi graph demonstrate that my homophily rearrangement algorithms work on networks with an arbitrary topology. A simple network which can be used during the sensitivity analysis is a square lattice with periodic conditions. When I analyze the sensitivity of the expected solution time to the distribution of the generic vertex feature and the size of the system, I use the above mentioned lattice topology with von Neumann neighborhood – so all of the vertices have 4 neighbors in the investigated networks. The R scripts that was used for the simulations is attached as Script E.13 in Appendix E.

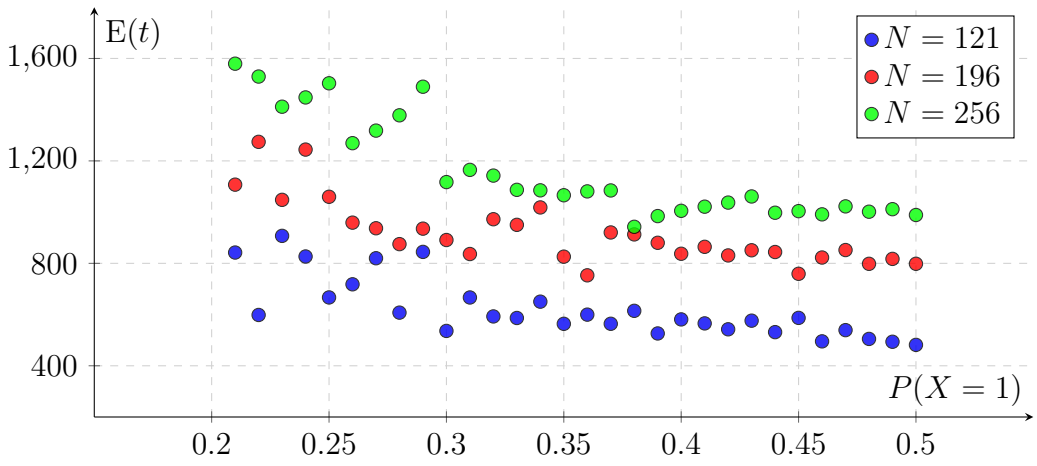


Figure 7.2: Expected convergence time of the heuristic algorithm as a function of system size and balancedness of the feature distribution

The simulation results regarding the effect of system size and initial feature distribution are summarized in Figure 7.2. The simulations ran on square lattices with 121, 196 and 256 vertices. The nodes had a generic vertex feature which had a Bernoulli distribution. The outcomes were either one or zero and the parameter of this distribution was varying between 0.21 and 0.5 with a step size of 0.01. Each of the points in Figure 7.2 represent the average of 200 convergence times. Actually a given point of the scatter plot represents the expected convergence time as a function of the system size and the ratio of agents with feature value equal to one. The horizontal axis marks the changing parameter of the Bernoulli distribution, the vertical axis denotes the expected convergence time and the coloring corresponds to the system size (number of vertices).

Based on the visualization, one can infer the existence of two important relationships. First, as the number of vertices increases, the expected convergence time also increases. This is not a surprising regularity, the feature value exchanges have atomic effects on the overall feature similarity on a large network, while on small networks the effect can be large. Nonetheless, this observed phenomenon can be relaxed if the degree distribution is not as particular as in case of a square lattice. Second, the balancedness of the feature distribution (the ratio of feature values equal to one and zero) decreased the expected convergence time. In Figure 7.2 only those particular cases are shown where the ratio of vertices with a feature equal to one is at 0.5 or below it. However, it is logical that the expected convergence time increases when the ratio of feature values equals to one is above 0.5. Generally, finding vertices that move the actual level of homophily towards the target is becoming harder.

Target homophily level

If the observed generic vertex feature has variance and it was assigned randomly to the vertices, initially the network shows only weak signs of homophily or heterophily. Fundamentally, it is an unordered state, from this unordered state the homophily rearrangement algorithms move the network into an ordered state – vertices mainly have neighbors that are similar or neighbors that are primarily dissimilar. It is an undeniable orderliness that achieving a state with high order is computationally costly. So it is expected that the number of steps needed for attaining a given level of homophily increases in the target homophily level. The simulation results that I will present demonstrate that this relationship exists. An R implementation of the experiment is enclosed in Appendix E as Script E.14.

The experiments and simulations were implemented for square lattices with periodic boundary conditions. The number of vertices in the square lattice was 100 – the size of the lattice was simply (10×10). During the simulations the single generic vertex feature has a Bernoulli distribution with a parameter equal to 0.5. The level of homophily is calculated with a universal homophily measure which gives a scalar measure of homophily (Moran's I). The scatter plots in Figures 7.3 and 7.4 were generated based on the following experiment. The initial assignment of the feature values to the vertices is random. The level of prescribed homophily varies between -0.5 and 0.5 with a stepsize of 0.01. For each level of homophily, I solve the homophily rearrangement problem 200 times, and store the average (expected) and median solution time. The algorithms used for solving the

homophily rearrangement problem are the heuristic and the greedy one.

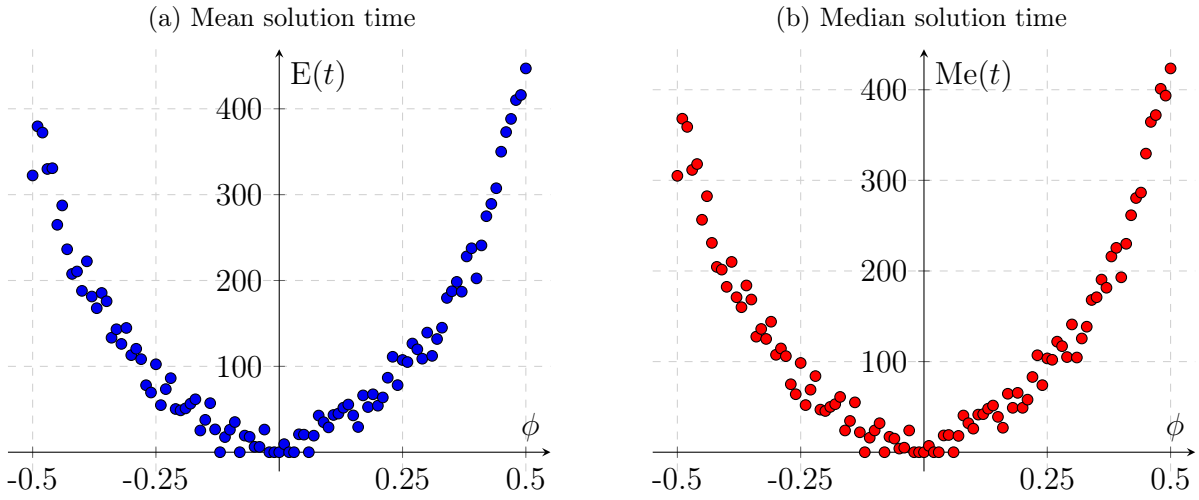


Figure 7.3: Expected average and median convergence times of the heuristic algorithm on a square lattice with periodic boundary conditions as a function of target homophily

The simulation results obtained with the use of the heuristic algorithm are plotted in Figure 7.3. The horizontal axes is the target homophily and heterophily level within the predefined interval, the vertical axes are the expected and median solution times respectively. One can observe that the solution time increases in the absolute level of heterophily or homophily. In addition, the relationship is non-linear, based on the plot it seems to be quadratic both for the expected and median solution times. In plain words, this means that the higher the level of order that we want to achieve in the system, the higher number of iterative steps we will need. The observed non-linearity is a byproduct of the random vertex pair selection during the feature value exchange process. Initially, finding vertex pairs that increase the homophily in absolute terms is a simple task, later as the assignment of features becomes ordered it is becoming a more cumbersome challenge. Plainly, randomly choosing a suitable pair of vertices becomes complicated.

If one looks at Figure 7.4, it turns out that the greedy algorithm needs a lower number of iterative steps to solve the same problem. The expected and median solution times of the algorithm are lower in case of the greedy algorithm. Moreover, the concave profile of the homophily – solution time curve is also different. It seems that the expected and median solution time is linear (with some random noise) in the target homophily value. This observed linearity is a consequence of the square lattice with periodic boundary conditions topology. The greedy algorithm chooses vertices that are in the neighborhood

of vertices that have 4 neighbors with dissimilar feature values. Because of this, each of the feature value exchanges has the same *unit* effect on the increase of the homophily. Similarly, when the target of the rearrangement is heterophily the feature value exchange takes place between a node which has similar neighbors and one which has dissimilar neighbors in terms of the generic vertex feature. The observed random noise is a result of the initial random assignment of the feature values to the nodes.

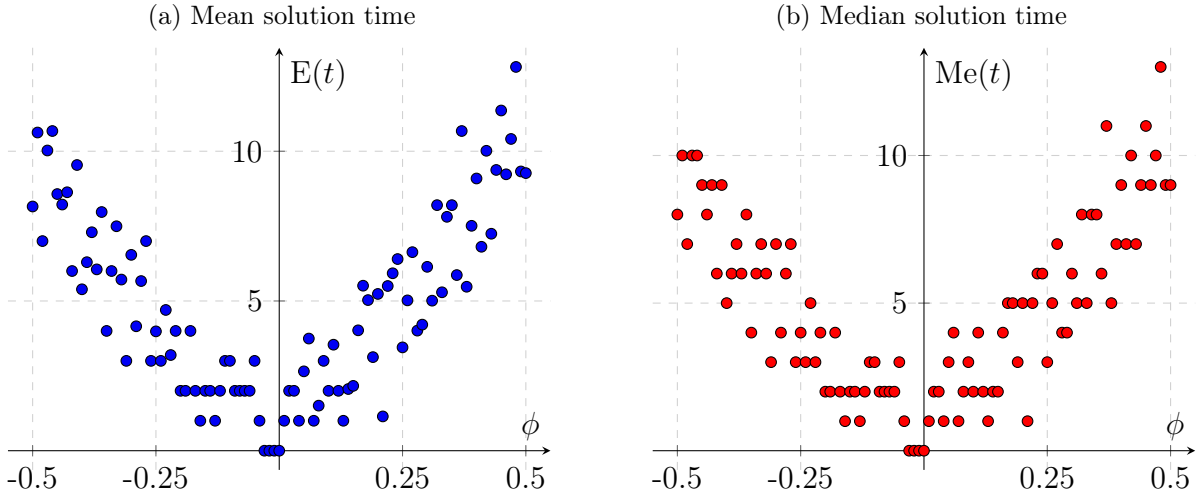


Figure 7.4: Expected average and median convergence times of the greedy algorithm on a square lattice with periodic boundary conditions as a function of target homophily

The difference of the mean and median solution times gives valuable information about the skewness of the solution time distributions as a function of the homophily objective values. These differences are plotted as a function of the target homophily value in Figure 7.5. The results obtained with applying the heuristic algorithm are shown in Subfigure 7.5a, while results generated by the greedy one can be seen in Subfigure 7.5b. In case of the heuristic homophily rearrangement algorithm, the mean-median difference is increasing in the homophily objective value. This means that the solution time distribution is becoming skewed and starts to have a tail on the right as the absolute level of homophily or heterophily increases (the mean is increasingly above the median). Moreover, the difference is increasing in a non-linear way – the shape of the curve is again quadratic. However, this is not true for the greedy algorithm. In Subfigure 7.5b, one can observe that the difference between mean and median is becoming more and more dispersed, but there is no clear linear or non-linear relationship.

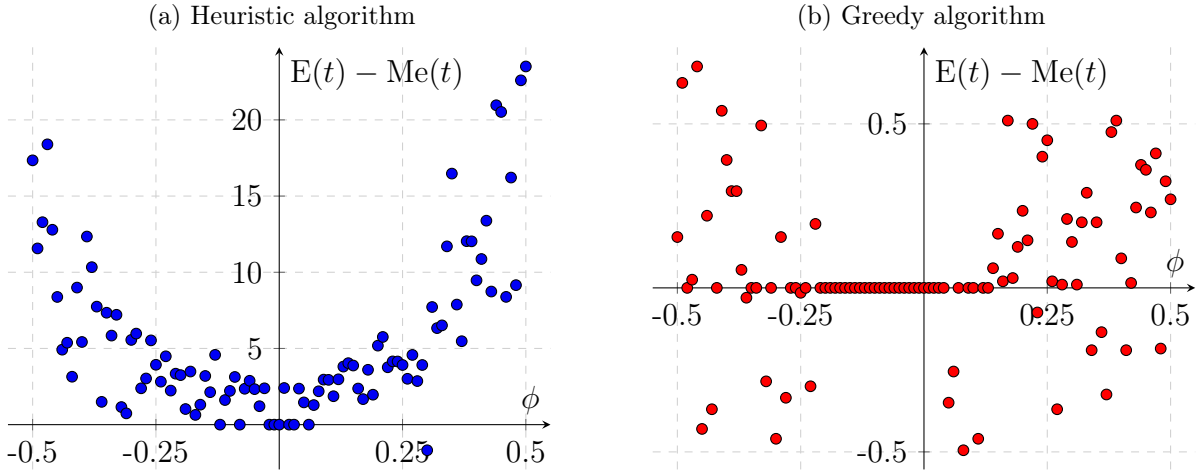


Figure 7.5: Difference in the mean and median convergence times as a function of homophily target value on a square lattice with periodic boundary conditions

7.1.2 Stability of solutions

The heuristic homophily rearrangement and its constrained variant (with bag of indices) choose the vertices that participate in the feature value exchange randomly. Henceforth, it is plausible that the homophily rearranged feature vectors that result from separate simulation runs with the same target value are different from each other. This results in two paramount phenomena that are worth further examination. One of them is the homophily inertia and the instability of solution times. The other one is the weak similarity of the resulting rearranged feature vectors. Particularly, that the rearranged generic vertex feature vectors that result from multiple simulation runs are just slightly correlated with each other and the original feature vector. In addition, it also holds that topological vertex properties and the rearranged generic vertex features are marginally associated.

It is straightforward that square lattices have a degree distribution which promotes the existence of unstable homophily rearrangement solutions. On the other hand, it is not a pronounced regularity when the network has a structure that is not as simplistic as it is in case of a square lattice. During the analysis about the stability of homophily rearrangements, I use a variant of the *National Longitudinal Study of Adolescent Health* (Harris, 2009). This is a dataset about friendship networks in high schools, which is used in a number of empirical studies about social networks (Moody, 2001; Gonzalez et al., 2007; Currarini et al., 2009). The specific focus of my analysis is the network of friendships obtained from school Nr. 81. Importantly, the network has a non-uniform

degree distribution described by a power-law, so it is remarkably different from a square lattice. As a preliminary data handling procedure, I removed students who had missing generic vertex features (grade, race and gender) or generic vertex feature values that are not meaningful. The number of nodes was 1268, and the number of edges was 4524. Distribution of the gender and grade variables is fairly uniform, this is why I focus later on these two generic vertex features. The gender feature is binary, while grade is ordinal with 4 different outcomes.

Homophily inertia and instability of the solution time

The term *homophily inertia* used by Yavas & Yusel (2014) describes the phenomenon that when a homophily rearrangement or homophilous network generation process starts certain groups end up with higher categorical homophily level than the target homophily level. The reason behind this is simple. If one of the groups reaches the target homophily level while others not, the group that reached the target level starts to show increased levels of homophily well beyond the target level. This means that the actual level of categorical homophily that a groups shows diverges from the target level. This phenomenon can be demonstrated with a few simulation runs.

First, I investigate homophily rearrangement regarding the gender variable. The chosen algorithm is the heuristic homophily rearrangement algorithm and I applied the inbreeding homophily index as a categorical homophily measurement function. Before the homophily rearrangement started, I randomized the features. In Figure 7.6 I plotted the level of inbreeding homophily in each iterative step regarding the gender variable in the high school. The actual level of homophily can be seen both for the males and females in the high school. The prescribed target level of homophily was 0.05 for both males and females. The unrelaxed version of the algorithm by design usually results in time series, where one time series is always above the other one because in each iterative steps both elements of the objective value vector must increase.

The first simulation run results in an inbreeding homophily of 0.07 for males and an inbreeding homophily of females around 0.05, for details see Subfigure 7.6a. The male specific inbreeding homophily is always above the female specific one. The convergence approximately took 5000 iterative steps. Another simulation's results are plotted in Subfigure 7.6b. In this run the initial homophily was higher for females, the resulting homophily was around 0.05 for both groups and the specific time needed for achieving the

preset level of homophily was 3500 steps. These two runs had shown that the inbreeding homophily levels can diverge and also that the time needed for achieving the objective value varies. This is not surprising due to the probabilistic nature of the heuristic algorithm.

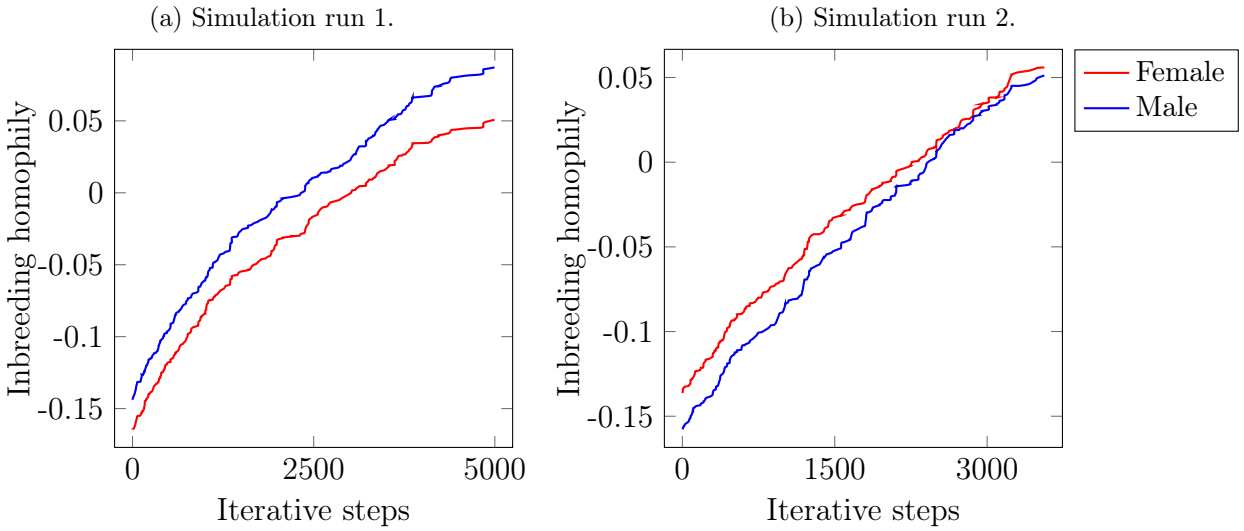


Figure 7.6: The convergence of gender based inbreeding homophily to a target vector in two separate simulation runs – based on the school friendship network

Second, regarding the grade variable I ran simulations to achieve a prescribed level of heterophily, for all the groups I wanted to have an inbreeding heterophily of -0.15. The algorithm used for obtaining a network that satisfies the heterophily condition was the heuristic one with relaxed switching conditions. Results of two distinct simulation runs are shown in Figure 7.7. The inbreeding homophily as a function of iterative steps is plotted in Subfigure 7.7a for the first simulation run. Due to the relaxation, flat regions of inbreeding homophily appear, for certain groups the homophily is increased while for others it is unchanged . In case of 9^{th} graders this is quite prevalent, during iterative steps between steps 250 an 500 the inbreeding homophily stops to decrease multiple times for longer periods. Also, one observes a divergence in the inbreeding homophily values just as before in case of the gender vertex feature. The 9^{th} graders had an objective value just below -0.15, while the 10^{th} graders had a value that was well below -0.23. The difference of in inbreeding homophily among groups in the solution is quite stark, the fact that there is no upper limit on inbreeding homophily results in divergence. The results of a second simulation run are plotted in Subfigure 7.7b. The flat regions in the level of homophily show that for certain random feature switches the vector which describes the homophily

values is essentially the same. For this second run, the time needed for achieving the prescribed homophily vector is considerably lower – it was approximately 270 steps.

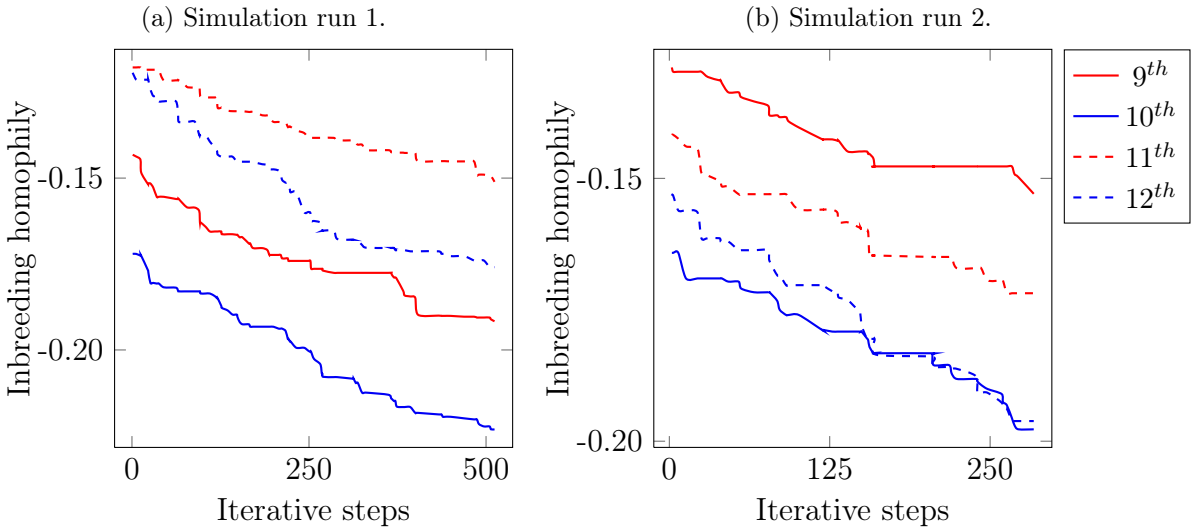


Figure 7.7: The convergence of grade based inbreeding homophily to a target vector in two separate simulation runs – based on the school friendship network

As the simulation results in Figures 7.6 and 7.7 show, the algorithm design results in *homophily inertia* – which was described by Yavas & Yusel (2014). The group specific homophily target values result in an average of group specific homophily (heterophily) levels that is above the average of the prescribed levels. This is even true, when the feature value switching condition was relaxed and steps that result in the same homophily or heterophily for a certain group were allowed. It should be emphasized that this phenomenon is only present when the feature of interest is categorical and the function used for measuring the homophily is a categorical homophily measurement function. The different simulation runs show that the strength of homophily inertia is different in multiple runs.

Stability of solutions

Results on the gender and grade based homophily rearrangement show the resulting features are distinct regarding homophily. In different runs, the same vertex might end up with different feature values. In the end, it results in different group specific homophily levels that are all at above a given level of homophily or heterophily as Figures 7.6 and 7.7 had shown. The instability of solutions can be proved by an uncomplicated experiment. The universal homophily regarding the original gender feature is approximately 0.27

(measured by Moran's I). Not surprisingly, students are more likely to have friends who have the same gender. With this information and with knowing the original ratio of males and females in the dataset, using the heuristic algorithm I generated 100 artificial gender features which satisfy that the homophily measured by Moran's I is roughly 0.27. A simple linear correlation can be calculated between each of these artificial features and the resulting correlation matrix is plotted in Subfigure 7.8a of Figure 7.8. None of the pairwise correlations is significant, which is quite surprising, considering that I tested the significance of 4851 pairwise correlations. This means that the resulting synthetic gender feature vectors are not similar to each other. The above mentioned phenomenon is even present even when the level of target homophily is increased. This idea is supported by the same simple experiment, the generation of 100 artificial gender feature vectors that correspond to a homophily level of 0.5. Later these features are correlated with each other. The resulting correlation matrix is visualized in Subfigure 7.8b, where the lack of strong positive or negative associations is visible.

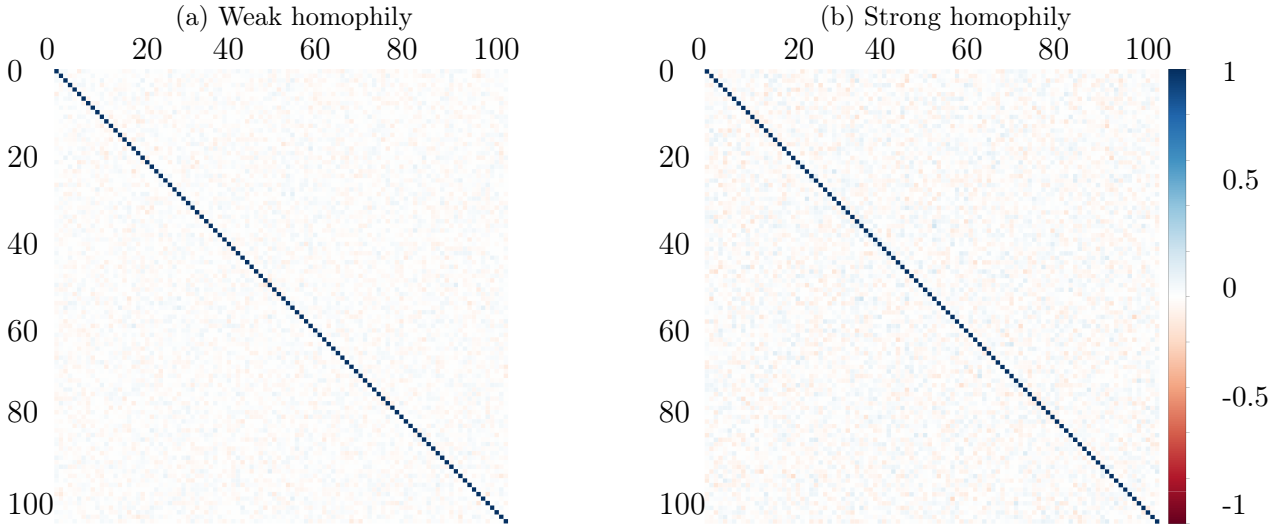


Figure 7.8: Linear correlations of the resulting feature vectors that show a target gender based homophily

The artificial generic and topological vertex features (betweenness centrality, degree or coreness) are possibly correlated with each other. Furthermore, these associations are likely to be influenced by the homophily or heterophily present on the network. With the proposed algorithms it is feasible to generate an arbitrary number of artificial features that satisfy a given homophily criteria. For a given pair of artificial generic and topological vertex feature a linear correlation coefficient is computable. This can be done for all

the simulated artificial features and the resulting correlation coefficients can be used for estimating the distribution of correlation among a given generic and topological vertex feature, while the generic vertex feature's distribution, network topology and homophily regarding the feature are fixed. The resulting distributions can be practical for making quite complex inferences about the actual correlations of the generic vertex feature and the topological features.

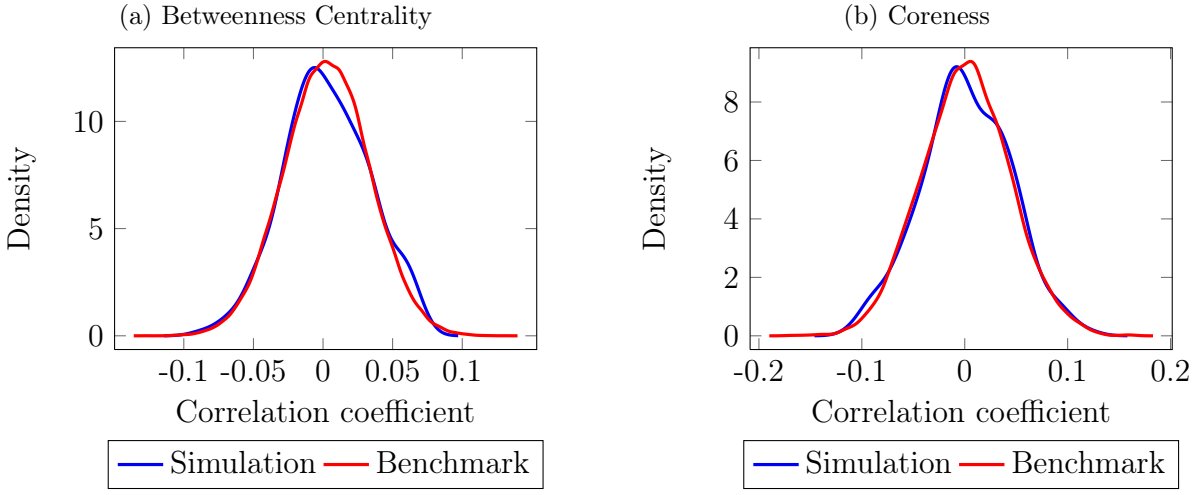


Figure 7.9: The distribution of correlation between a homophily rearranged generic vertex feature and topological properties based on the school friendship network

Panels of Figure 7.9 show the distribution of the correlation coefficients among topological vertex features and replicated artificial gender features – these are the feature vectors that were utilized for creating Subfigure 7.8a. Densities of the distributions were estimated from 1000 synthetic feature replicates with a kernel density estimator. The target homophily was set to be 0.27 again. The kernel used Gaussian smoothing and the bandwidth of the kernel was set according to recommendations by Silverman (1986). The R code snippet that generated the simulation data is attached in Appendix E as Script E.15.

In Subfigure 7.9a, the correlation between gender and the betweenness centrality of vertices. The distribution of linear correlation coefficients is unimodal and compared to a normal distribution (with the same expected value and standard deviation) it is skewed slightly. The actual level of correlation between the original gender variables and between centrality was 0.05. Based on the simulation results, it comes that this factual correlation of betweenness centrality and gender is higher than 91.9% of the simulated correlations that were obtained by generating synthetic gender features and correlating them with the

betweenness centrality. The distribution of the correlations among gender and coreness is shown in Figure 7.9b, it is inferable that the distribution is seemingly close the the normal, but compared to a normal distribution which has the same expected value and standard deviation, it is weakly skewed. Importantly the low number of generated observations might influence these findings. The factual level of linear correlation between coreness and gender was 0.13. Based on the simulation results, it follows that the observed correlation of coreness (degeneracy) and gender is higher than 97.3% of the simulated correlations that were obtained by generating synthetic gender features and correlating them with the coreness vector. As these two examples has shown, sophisticated inferences can be formed from the results of the algorithms, as in both cases the topology of the network is unchanged, the feature's distribution is the same and the homophily level is approximately identical.

These simulations regarding the gender and grade variable support the existence of three phenomena. First, the number of iterative steps needed to achieve a a given level of homophily or heterophily is volatile – the initial assignment of feature values influences the convergence time. This is is also supported by the sensitivity analysis results. Second, there is evidence of homophily inertia regarding binary and ordinal (categorical) features. Third, the resulting feature vectors do not show associations with each other – there are multiple solutions that have the same level of homophily.

7.2 Multivariate simulations

The three proposed homophily rearrangement algorithms were generalized to multivariate networks in Chapter 5. First of all, my results demonstrate that these algorithms are capable of reshuffling multiple generic vertex features in order to create networks that show homophily regarding more than one dimension. The proposed algorithms have similar properties as the univariate homophily rearrangement algorithms with respect to sensitivity (in terms of target homophily and system size). In the remainder of the section I show with simulation results three important properties of the heuristic multivariate homophily rearrangement algorithm. First, I show that the expected convergence time to the target homophily level depends on the correlation of the generic vertex features in Subsection 7.2.1. Second, I establish in Subsection 7.2.2 that the expected solution time increases with the system size. Moreover, the product of the experiments in Subsection 7.2.2 implies that the expected solution time is quadratic in the correlation of the

generic vertex features. Third, in Subsection 7.2.3 the simulation results expose that the homophily level increases sub-linearly with the number of iterative steps. In all of the experiments I assume that the features are continuous and the homophily regarding the variable is measured by Moran's I metric.

7.2.1 Sensitivity to the generic vertex feature correlations

The introduction of multiple generic vertex features in the system allows for correlations among the features. The observed level of correlation among the generic features influences the convergence to the target homophily level. It is evident that if two generic vertex features are perfectly correlated, the multivariate nature of the system will not influence the expected convergence time. However, in most of the real networks, the generic vertex features are only slightly correlated with each other. To show that the correlation (and absence of correlation) among the vertex features influences this convergence time, I implemented homophily rearrangement experiments.

In all of the experiments I used the multivariate heuristic homophily rearrangement algorithm on square lattices. The size of the square lattice was fixed at 10×10, so the number of vertices was 100 in the system. The lattice had periodic boundary conditions, so every single vertex had connection to 4 other vertices. The nodes had two generic vertex features, denoted by $\mathbf{x}$ and $\mathbf{y}$. These generic vertex features had standard normal distributions and the correlation between them was described by the parameter ρ. Altogether I implemented 6×1000 separate simulation runs, the parameters that I adjusted were the correlation between the features and the target homophily level. Correlation between the generic vertex features was described by three different states: negative correlation ($\rho = -0.5$), the lack of correlation ($\rho = 0$) and positive correlation ($\rho = 0.5$). The target homophily level was the same for the two features. In one case the target was homophily $\Phi = (0.5, 0.5)$ in the other it was heterophily with respect to both variables, so the target vector was $\Phi = (-0.5, -0.5)$. The R implementation of the experimental setting and the functions generating the plots are attached in Appendix E as Scripts E.16 and E.17. The distribution of the resulting convergence times to homophilous or heterophilous states is plotted in Figure 7.10.

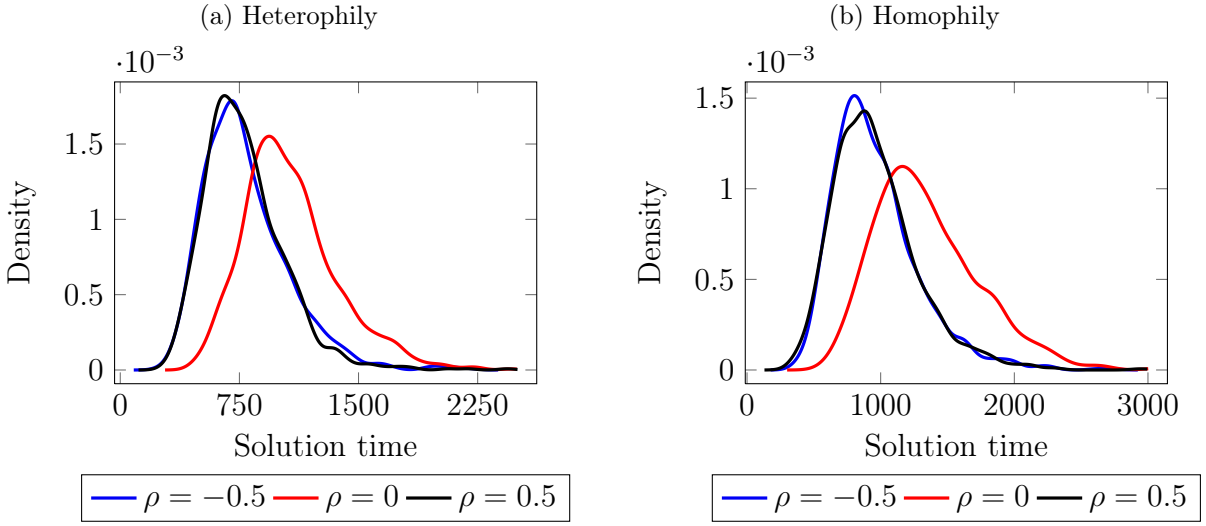


Figure 7.10: The distribution of expected solution times conditional on the correlation of generic vertex features

In Subfigure 7.10a one can see the distribution of solution times obtained when the target of the multivariate homophily rearrangement was heterophily for both variables. It is evident that both positive and negative autocorrelation reduces the expected (mean) convergence time of the heuristic homophily rearrangement algorithm. Interestingly, the solution time distributions obtained from the runs when generic vertex feature correlation was present are closely overlapping.[1] The distribution of the solution times when the target was heterophily is shown in Subfigure 7.10b. This also supports that solution times were lower on average when the features were correlated with each other. My later results will support that the increase in correlation results in lower expected and median solution times. Moreover, it will demonstrate that the effect on the solution time is non-linear.

7.2.2 Sensitivity to the system size

Univariate simulations indicated that as the system size is increasing, the time needed to rearrange the feature in a homophilous ways is also increasing. As I argued, this comes from the fact that the effect of a single vertex feature exchange is becoming atomic, the effect on the global similarity of features becomes negligible. The same phenomenon can be validated regarding multivariate homophily rearrangement, namely that the number of iterative steps increases with the number of vertices in the network. In the previous

[1]Based on the respective Kolmogorov–Smirnov, test one could conclude that the observed data points are drawn from the same theoretical distribution.

subsection I have only established that the correlation of features reduces the time needed for solving the homophily rearrangement problem. However, I did not specify the exact relationship between the expected solution time and the correlation of generic vertex features. With the proper design of experiments both of the above mentioned regularities can be proven.

The multivariate homophily rearrangement simulations were implemented on square lattices which had varying size. The correlation among the generic vertex features also varied. The respective sizes of the lattices were between 10×10 and 16×16, which means that in the largest network the number of vertices was more than double the number of vertices in the smallest network. The vertices had two generic vertex features, denoted by $\mathbf{x}$ and $\mathbf{y}$. These generic vertex features had a standard normal distribution and the linear correlation of the features was between -0.5 and 0.5 with a stepsize of 0.1. For each parametric set up (network size and correlation between features) I ran 1000 simulations. The R script used for the simulation set up is attached in Appendix E as Script E.18.

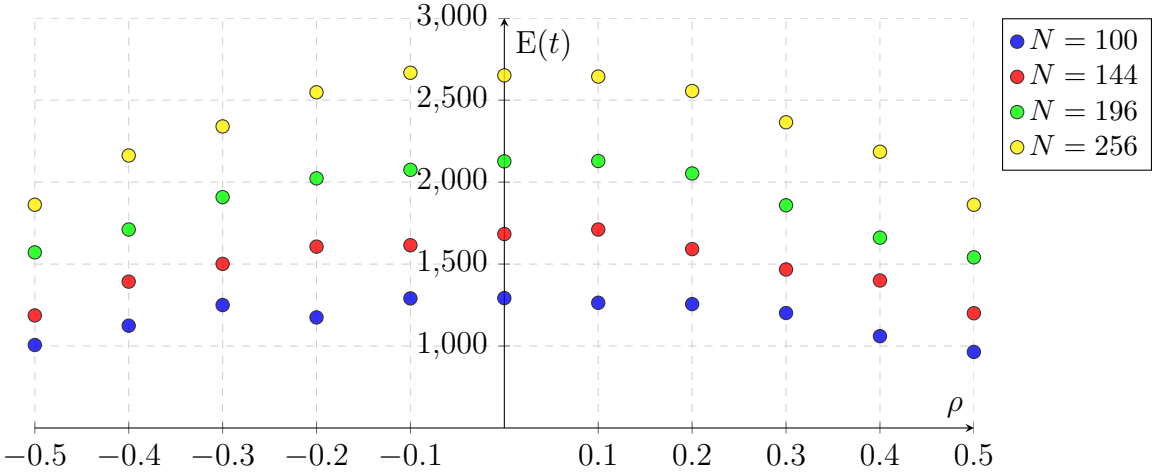


Figure 7.11: Expected solution time of the multivariate heuristic homophily rearrangement algorithms as a function of feature correlation and lattice size

In Figure 7.11, I plotted the mean solution time obtained from the simulation runs as a function of the generic vertex feature correlation (horizontal line) and the system size (color of the dots). Each of the points on the scatter plot represents the mean of 1000 simulation runs. There are two phenomena that can be inferred with high confidence based on the simulation results. First, the average time needed for solving the homophily rearrangement problem increases with the size of the network. Second, as correlation increases in absolute terms the time needed for solving the homophily rearrangement

problem is also increasing. Moreover, the scatter plot in Figure 7.11 supports that the expected solution time is a quadratic function of the absolute correlation time. The median of the solution times is plotted on Figure D.2 in Appendix D. The medians also support that the solution time depends positively on the system size and negatively on the absolute level of feature correlation.

7.2.3 Convergence to the target homophily level

Simulation results in Subsections 7.2.1 and 7.2.2 show that the expected convergence time depends on the correlation of the generic vertex features. Not only the expected convergence time to a homophilous state, but the expected level of homophily during the iterative process at a given time period also depends on the correlation of the features. This simple regularity about the faster convergence to a homophilous or heterophilous state conditional on correlated features can be underpinned by a set of experimental simulations on square lattices. Moreover, my results imply that the correlation of the generic vertex features mitigates homophily inertia – the homophily levels diverge from each other less when the features are correlated.

In order to demonstrate the existence of the above mentioned regularity, I set up an experiment where the topology of the networks was defined by a square lattice with periodic boundary conditions. The size of the lattice was 10×10, and the vertices had 2 generic vertex features which were standard normally distributed. In order to investigate the effect of homophily, I investigated three different cases regarding the correlation: negative correlation of the features, no correlation and positive correlation between them. The feature values were assigned randomly to the vertices, only the correlation between them was preset. There were separate runs with homophilous targets and heterophilous targets, the respective homophily target vectors were $\Phi = (0.5, 0.5)$ and $\Phi = (-0.5, -0.5)$. For each scenario (preset correlation level and homophily target value), I generated 1000 multivariate homophily rearrangement simulations and saved the time specific homophily or heterophily levels. Time specific homophily levels are averaged out for the simulations and the two features. This simply means that each point is a the average of 2000 homophily levels for a given time period. The R implementation of the respective experiments can be found in Appendix E as Scripts E.19, E.20, E.21 and E.22.

In the subfigures of Figure 7.12, I plotted the expected homophily and heterophily level as a function of time. Based on the results summarized by Subfigure 7.12a, it is evident that the expected time dependent homophily level is the same when the features are correlated positively or negatively with each other and the correlations equal in absolute terms. The observed homophily level is always lower when the the features are uncorrelated with each other – it is not surprising considering that the correlation of the features speeds up the convergence. The same conclusion holds when one looks at the simulation results obtained with targets of heterophily. When features are uncorrelated, the convergence to the target state is somewhat slower. The reason behind this observed regularity is simple. If two features are correlated, then a given observation of feature values is likely to be similar to the neighboring nodes' feature or dissimilar. Therefore, both of them would likely to move the target homophily levels towards the same direction. On the other hand, when they are uncorrelated the similarity or dissimilarity from the neighbors' features is independent. On accounts of this, choosing suitable pairs of vertices for the feature value exchange becomes unlikely.

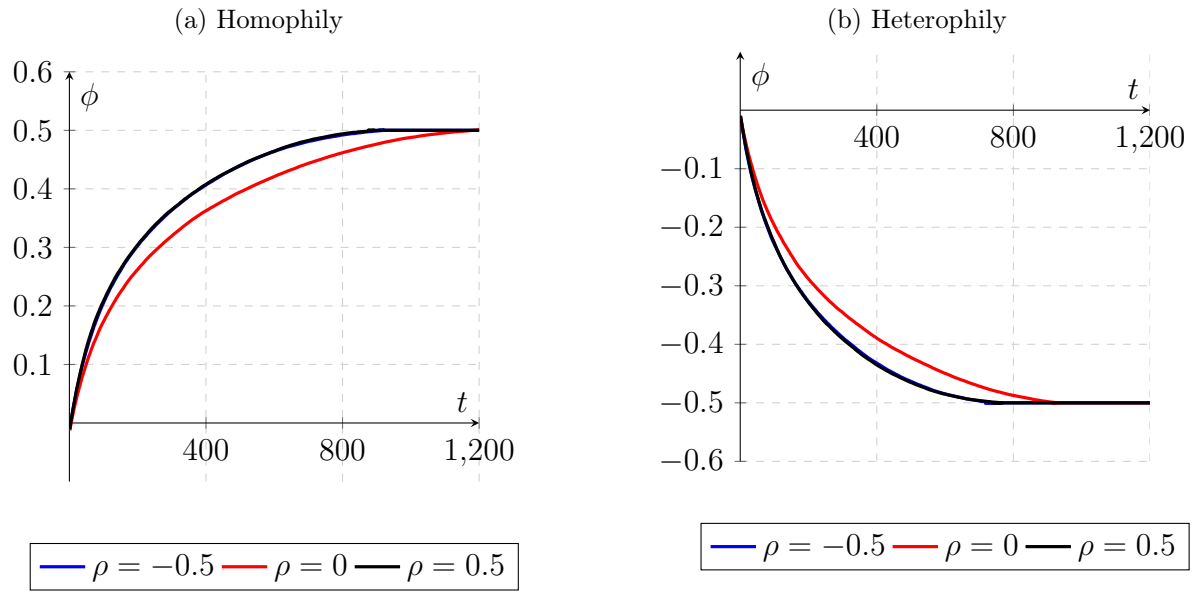


Figure 7.12: Expected level of homophily as a function of time and correlation among the generic vertex features

Based on the time dependent homophily levels simulated previously an expected absolute difference of homophily levels can be calculated by applying the following simple procedure. First, for each simulation run we calculate the time specific homophily levels for each of the variables. Second, we take the difference of them – this shows the time specific

difference in homophilies. Third, if one takes the absolute value of the differences, they can be averaged out and plotted as a function of time. This is the way how I obtained the time series in Figure 7.13.[2] First, one can observe that the absolute difference in homophily levels is decreasing with the number of iterations. Second, based on Subfigures 7.13a and 7.13b, it follows that the correlation of generic vertex features reduces homophily inertia – this is not an unpredictable fact. Let us image that two features would be strongly correlated, then the homophily levels regarding them would be also close to each other.

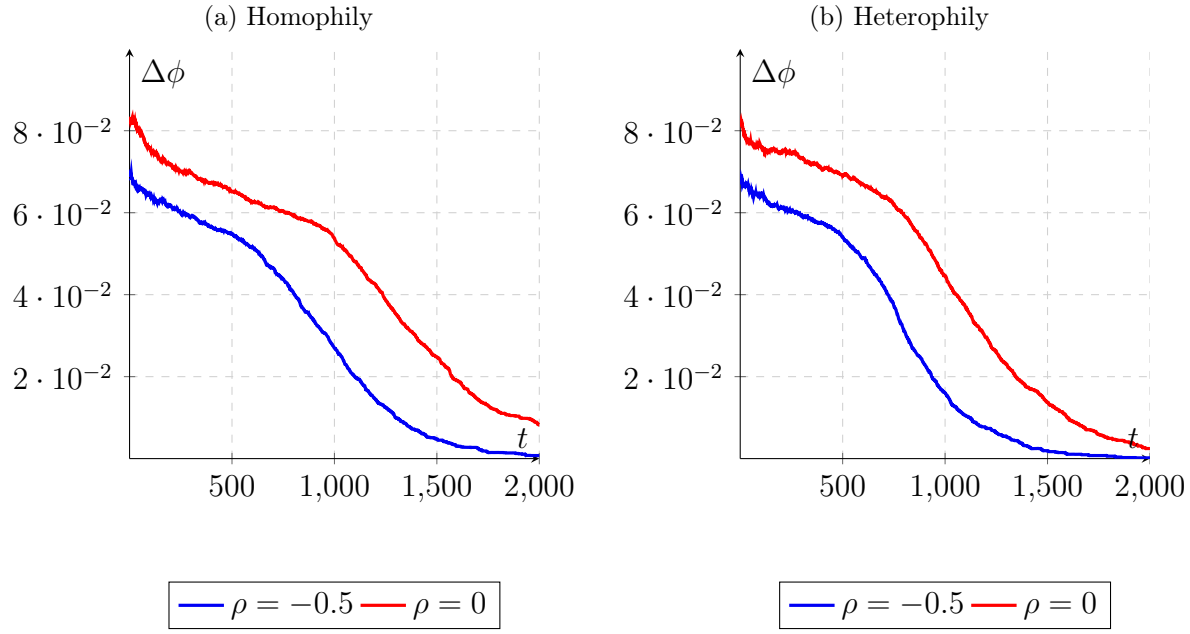


Figure 7.13: Expected absolute difference of homophily as a function of time and correlation among the generic vertex features

[2]Figures 7.12 and 7.13 were generated from the same simulation data.

Chapter 8: Simulation of diffusion

The proposed homophily rearrangement algorithms allow for controlled levels of homophily in networks that have an arbitrary topology. In Chapter 6, I proposed a model of diffusion where the transmission of an idea, information or infection depends on the similarity of the agents' generic vertex properties. As I highlighted previously, homophily is an aggregated measure of generic vertex feature similarity. By applying the homophily rearrangement algorithms, one can investigate the similarity based diffusion on a network where homophily of a certain feature (or multiple features) is under control. To put it simply, one can simulate similarity based diffusion on homophilous or heterophilous networks without changing the topology and the distribution of the generic vertex feature.

This chapter comprises two sections that present the simulation results obtained by applying the similarity based diffusion model. Section 8.1 focuses on the basic similarity based diffusion model and investigates the sensitivity of the model to the parameters. In Section 8.2, I analyze the effects of heterogeneous sensitivity to generic vertex feature dissimilarity.

8.1 Sensitivity analysis

The sensitivity analysis that I carry out in this section has three focal points. First, in Subsection 8.1.1, I examine the effect of homophily on the convergence time to a fully infected state. My goal is to demonstrate that the expected convergence time and the expected ratio of infected nodes in the system is affected by the level of homophily. Second, in Subsection 8.1.2, I analyze how the change in the baseline transmission probability effects the convergence of the similarity based diffusion model. Third, in Subsection 8.1.3, I run simulations to investigate the effect of changes in the sensitivity parameter to the spreading phenomenon.

8.1.1 Sensitivity to homophily

The pairwise transmission probability equation's dependence on the dissimilarity of agents accounts for the fact that homophily influences the spread of information on the network. This phenomenon, namely that changes in homophily influence the distribution of the pairwise transmission probability values is proved by the following experiment. Let us imagine a square lattice with periodic boundary conditions, the size of the lattice is 50×50, so the number of agents is 2500. The agents have a single generic vertex feature denoted by $\mathbf{x}$, which has a standard normal distribution. The transmission probability between agents i and j during my simulation is expressed by Equation (8.1). So the dissimilarity is expressed by the Manhattan distance of the feature values.

$$P_{i,j} = P_0 \cdot \exp\left(-\gamma \cdot |\mathbf{x}_i - \mathbf{x}_j|\right) \tag{8.1}$$

The black and blue distributions in Figure 8.1 were generated by rearranging $\mathbf{x}$ in homophilous and heterophilous way with a target Moran's I of 0.5 and -0.5. The algorithm used for the rearrangement was the heuristic one. The γ parameter was equal to 1, the baseline transmission probability P_0 was 0.45 and the base function was the exponential. As can be seen from Figure 8.1, homophily increases the transmission probabilities because the dissimilarity among agents is lower. Similarly, the heterophily decreases the transmission probability because the dissimilarity increases between the agents. From these it follows that the change in the transmission probabilities effects the convergence of the diffusion process. It is also evident that the shape of the distributions is changed by the feature value rearrangement. In case of a homophilous state, the distribution has a tail on the left, otherwise, it is less skewed.

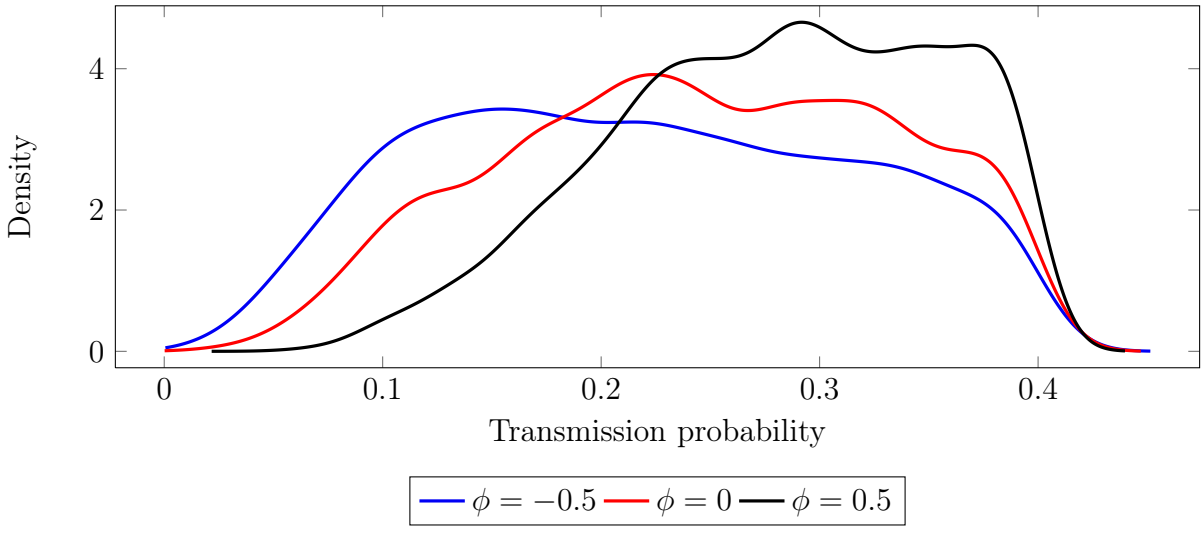


Figure 8.1: The distribution of pairwise transmission probabilities

The convergence to a state when all of the nodes are infected is effected by homophily through the increased pairwise transmission probabilities. This can be proven by an experiment where the topology is fixed, the distribution of the feature is fixed and the feature is rearranged in heterophilous and homophilous ways. The network used in the simulations has a square lattice topology with periodic boundary conditions and the size of the lattice was 10×10. Vertices have a single generic vertex feature $\mathbf{x}$, which has a standard normal distribution. The homophily was measured by Moran's I and the target values of the homophily rearrangement were 0.8 and -0.8 respectively. First, I generated a random assignment of the feature values. Second, I applied the heuristic homophily rearrangement 100 times and on the obtained networks that show the given levels of homophily. Third, I initiated a similarity based diffusion process with a single seeder. The pairwise transmission probability equation was defined by Equation (8.1). The baseline probability was 0.45 and the sensitivity to dissimilarity was one unit.

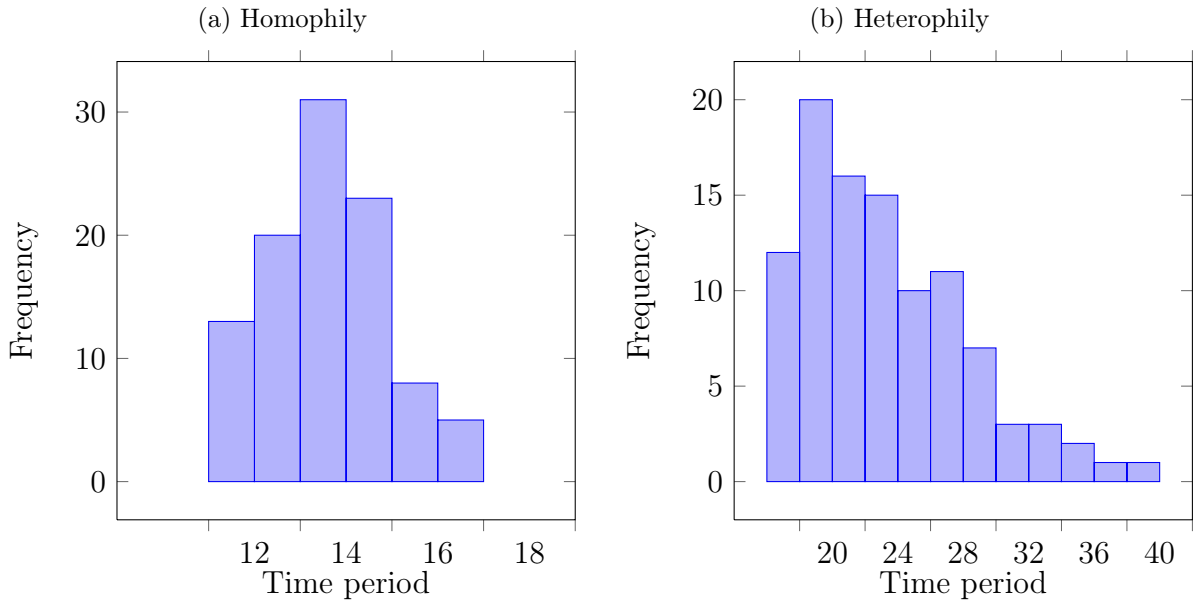


Figure 8.2: Distribution of the time needed for a perfectly infected state under homophily and heterophily

In Figure 8.2, I plotted the frequency of the solution times obtained from the simulation experiments. The results obtained with the homophilous networks are plotted in Subfigure 8.2a, while the results obtained under heterophily are plotted in Subfigure 8.2b. It can be inferred that the average solution time when homophily is present is lower. Moreover, the variance of the distribution is also lower compared to the the case when the network shows a strong heterophily regarding $\mathbf{x}$. The values plotted on the two histograms in

Figure 8.2 barely overlap with each other. This implies that the change in homophily influences heavily the convergence time to a fully infected state even when the network topology is fairly simple and the size of the network is small.

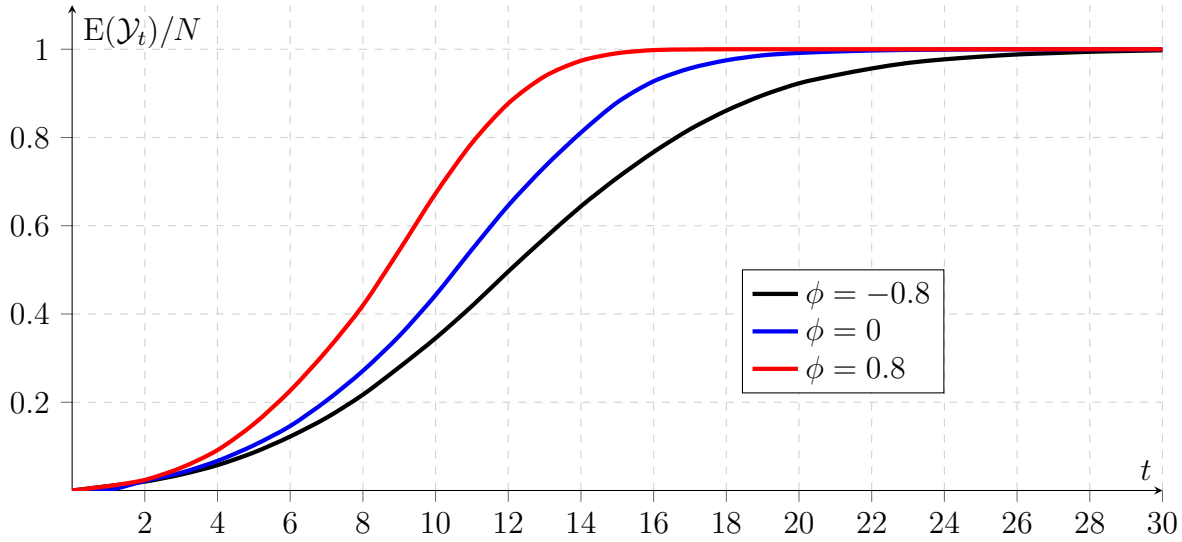


Figure 8.3: The ratio of infected nodes as a function of time

The simulation results that were used to generate Figure 8.2 can be used for highlighting another regularity of the similarity based diffusion model. Different levels of homophily result in different levels of mean ratio of infected nodes at a certain time period. For each level of homophily, an expected time specific ratio of infected nodes, here denoted by $\mathrm{E}(\mathcal{Y}_t)/N$, can be calculated by dividing the expected number of infected nodes at time period t by the total number of nodes. The curves in Figure 8.3 were obtained by doing the before mentioned operation on the simulation data that was created for Figure 8.2. The horizontal axis is the time ticker, while the expected ratio of infected nodes is noted on the vertical. It should be noted that the curves were smoothed out as the time ticker is discrete and one would only have data points at integer time points. As Figure 8.3 shows all of the curves have sigmoid shapes, which is not different from the results of the susceptible-infected model. Nevertheless, the main finding is that under homophily the time specific expected ratio of infected nodes is higher than under the lack of homophily. Similarly, under heterophily the time specific expected ratio of infected nodes is lower than under the lack of homophily. These findings also underpin that homophily helps the propagation of information, if one assumes that the information transmission probability depends negatively on the dissimilarity among nodes. The R scripts that were used for generating data and creating Figures 8.2 and 8.3 are attached in Appendix E as Scripts E.24 and E.25.

8.1.2 Sensitivity to the baseline transmission probability

A major tuning parameter of the similarity based diffusion model is the baseline transmission probability. The sensitivity of the model to this parameter is visible – if the baseline transmission probability increases, the convergence to a state where all of the nodes have the information becomes shorter. This connection between the baseline transmission probability and homophily is shown by my simulation results. In the first simulation, the network of interest is again a square lattice with boundary conditions. The size of the lattice is 50×50 and the vertices have a single generic vertex feature $\mathbf{x}$, which has a standard normal distribution. The steps implemented in order to generate the distributions in the subfigures of Figure 8.4 were as follows.

1. Random assignment of the feature values to the vertices. Homophilous and heterophilous rearrangement of the feature values. The respective target values of Moran's I were 0.8 and -0.8.

2. Calculating the pairwise transmission probabilities for the unordered, homophilous and heterophilous networks assuming that the pairwise transmission probability equation is is described Equation 8.1. The macro-level parameters are set such as $P_0 = 45$ and $\gamma = 0.5$.

3. Changing the baseline transmission probability to $P_0 = 0.9$ and recalculating the pairwise transmission probabilities.

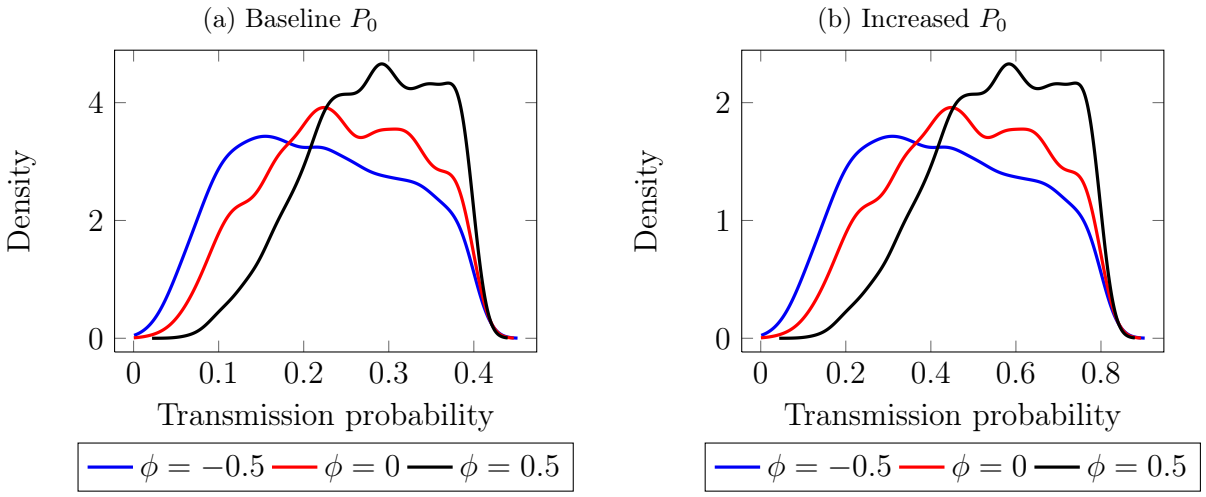


Figure 8.4: The effect of increased baseline transmission probability on the distribution of pairwise transmission probabilities

First of all, in Figure 8.4, one can observe that the change of the baseline transmission probability only rescales the distribution of the pairwise transmission probabilities. As one can see, the rescaling results in that the average and median of the pairwise transmission probability values among the vertices simply doubled by the doubling of the baseline transmission probability. The change in the transmission probabilities also affects the expected convergence time to a state when all of the nodes have the information (infected). This effect is quantifiable by averaging out simulation runs while one changes the baseline transmission probabilities and the homophily is also controllable with the heuristic homophily rearrangement algorithms. The network of interest had a square lattice topology, the size of the lattice was 10×10 and vertices had a single vertex feature. Simulations were implemented on a network that shows homophily, heterophily or one one which was not feature rearranged. For each homophily level and baseline probability combination, I simulated 200 similarity based diffusion processes. The homophily level was measured by Moran's I, the sensitivity to dissimilarity was equal to 1, and the pairwise transmission probability equation was described by Equation (8.1). The baseline transmission probability varied between 0.4 and 0.8 with a stepsize of 0.05. The respective R script is added in Appendix E as Script E.26.

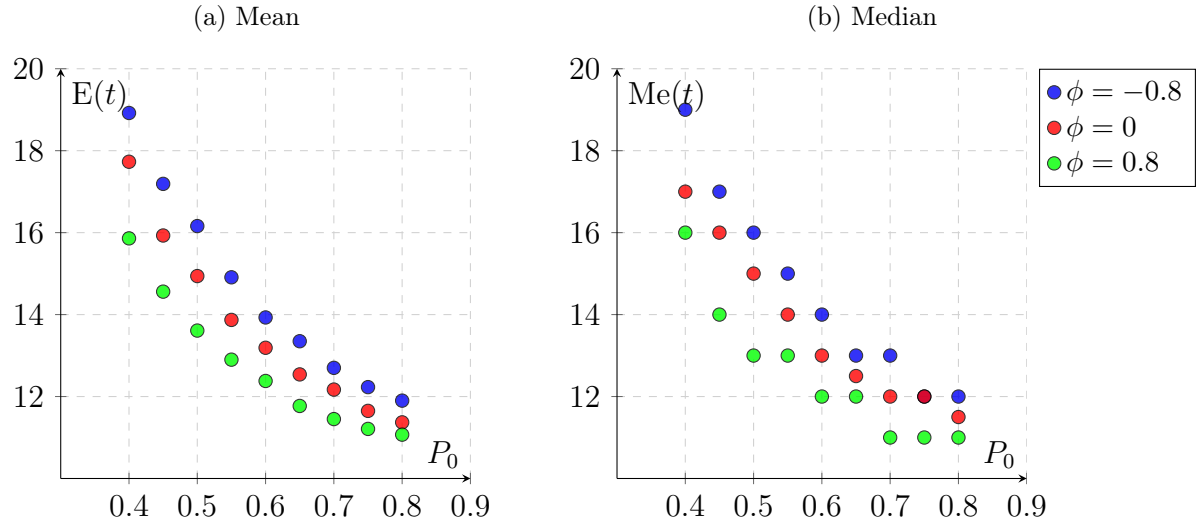


Figure 8.5: Expected and median solution time as a function of baseline transmission probability

In the subfigures of Figure 8.5 I plotted the mean and median convergence times to a fully infected state. One can observe that the increase in the baseline transmission probability results in decreased mean solution time. This is apparent from Subfigure 8.5a. Moreover, as it seems the effect of the transmission probability on the expected convergence time is

non-linear. Similarly, the results on Subfigure 8.5b imply that the increase of the baseline transmission probability decreases the median solution time. These results are in line with the changed distribution of pairwise transmission probabilities.

8.1.3 Sensitivity to the dissimilarity

Another major tuning parameter of the similarity based diffusion model is the sensitivity to dissimilarity. The intuition regarding the sensitivity to dissimilarity is that, when it increases the expected convergence time to a state where all of the nodes have the information also increases. This simple relationship again has roots in the changing distribution of the pairwise transmission probabilities. Let us consider a similar experiment to the one that was used for generating Figure 8.4. A square lattice with periodic boundary conditions, with a size of 50×50 and a single generic vertex feature, which has a standard normal distribution. In order to show the effect of the changed sensitivity to the pairwise transmission probabilities I constructed a simple simulation experiment. The steps of the simulation experiment were as follows.

1. The feature values are assigned randomly to the vertices, and the are rearranged in a homophilous and heterophilous manner. The target of these rearrangements was 0.8 and -0.8. The resulting rearranged feature values are the same as the ones used for generating Figure 8.4.

2. These rearranged feature values are sufficient for calculating the pairwise transmission probabilities between the vertices. The chosen pairwise transmission probability equation is defined by Equation 8.1. The parameters are chosen such as $P_0 = 45$ and $\gamma = 0.5$.

3. The sensitivity to dissimilarity is increased to be equal to 1 and the pairwise transmission probabilities are recalculated.

The distributions of the resulting pairwise transmission probabilities are plotted in Figure 8.6. The increased sensitivity to dissimilarity not just rescales the respective distributions, but makes them positively skewed. This skewness is the strongest when the network is heterophilous regarding the generic vertex feature of interest. Nonetheless, the shape of the distribution is changed even when the system shows homophily regarding the feature or when homophily is not present. This phenomenon foreshadows a later result, namely that the increased sensitivity increases the expected convergence time most on networks that show heterophily. The related R script is enclosed in Appendix E as Script E.23.

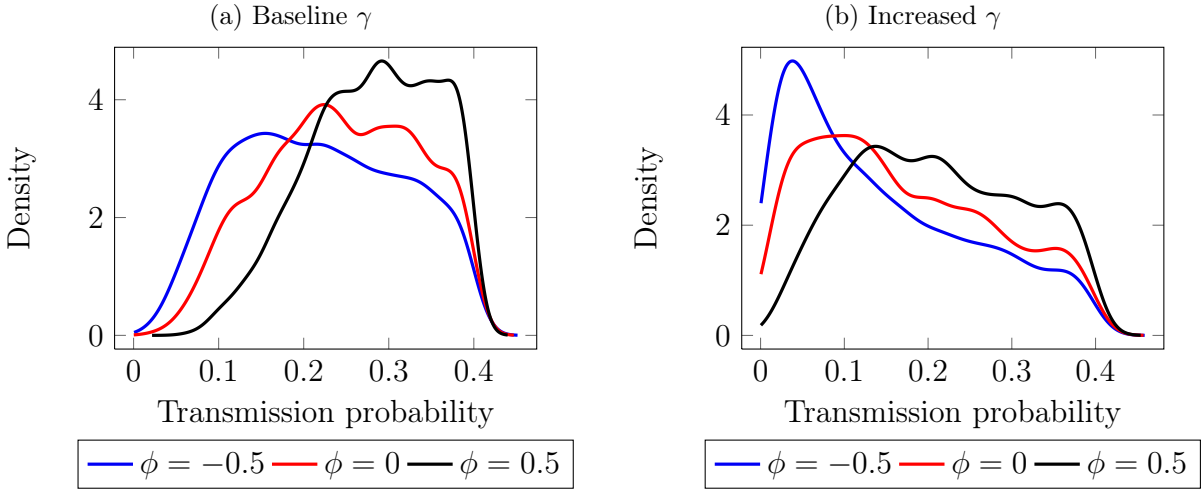


Figure 8.6: The effect of increased sensitivity to dissimilarity on the distribution of pairwise transmission probabilities

The nature relationship between the sensitivity to dissimilarity and the expected and median convergence times is assessable by similarity based diffusion simulations. The spreading process happens on a square lattice with periodic boundary conditions. The vertices have a single generic vertex feature, this feature has a standard normal distribution. Next, the feature is rearranged in order to generate networks that show homophily or heterophily. During the homophily rearrangements the homophily measurement function was Moran's I and the respective homophily rearrangement target values were 0.8 and -0.8. Initially, the generic vertex feature is assigned to the vertices in a random way.

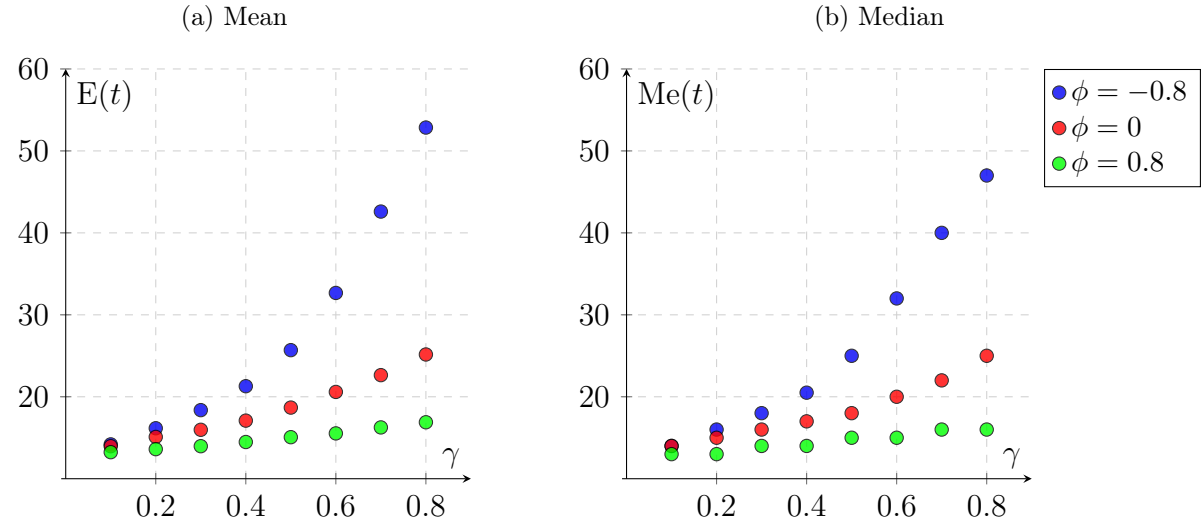


Figure 8.7: Expected and median solution time as a function of sensitivity to dissimilarity

The baseline transmission probability was 0.5 and the γ value varied between 0.1 and 0.8 with a stepsize of 0.1. The pairwise transmission probabilities were specifically defined by Equation 8.1. In Subfigure 8.7a of Figure 8.7, each point represents the expected convergence time of the diffusion process calculated from 200 simulations. It is evident that the expected convergence time is increasing in the sensitivity. Moreover, this increase is the highest when the features are rearranged in a heterophilous way. The median convergence time's behavior can be observed in Subfigure 8.7b, where values were again calculated from 200 simulation runs for each pair of parameters (homophily and sensitivity). The median solution time is also increasing in the sensitivity just as the expected solution time. This R implementation of the experiment is enclosed in Appendix E as Script E.27.

8.2 Heterogeneous sensitivity

The core idea of the similarity based diffusion model with heterogeneous sensitivity to dissimilarity is that the probability of passing on the information between two agents can be different. In such settings, when the dissimilarity is weighted heterogeneously, the exact effect of homophily will depend on the breed of the initial seeder node. In this section I demonstrate this regularity with similarity based diffusion simulations.

In my simulations, the system of interest is a network, which has square lattice topology with periodic boundary conditions. The nodes have a single generic vertex feature $\mathbf{x}$, and this variable is binary with two numeric outcomes. The probability that the outcome is 1 equals to 0.3 and, from this, it comes that the the probability that the outcome is 0 equals to 0.7. In the non homophilous set up of the network, the feature values are assigned randomly to the vertices. The pairwise transmission probability equations between nodes i and j are described by Equations 8.2 and 8.3.

$$P_{i,j} = P_0 \cdot \exp\left(-\gamma_1 \cdot |\mathbf{x}_i - \mathbf{x}_j|\right), \text{ if } \mathbf{x}_i = 1 \tag{8.2}$$

$$P_{i,j} = P_0 \cdot \exp\left(-\gamma_2 \cdot |\mathbf{x}_i - \mathbf{x}_j|\right), \text{ if } \mathbf{x}_i = 0 \tag{8.3}$$

From this it comes that there are three types of pairwise transmission probability values in the system. First, if $\mathbf{x}_i = \mathbf{x}_j$, then it holds that $P_{i,j} = P_{j,i} = P_0$. Second, if $\mathbf{x}_i = 1$ and $\mathbf{x}_j = 0$, then it comes that $P_{i,j} = P_0 \cdot \exp(-\gamma_1)$. Third, if $\mathbf{x}_i = 0$ and $\mathbf{x}_j = 1$, then one knows that $P_{i,j} = P_0 \cdot \exp(-\gamma_2)$. If γ_1 is larger then γ_2, then agents who have $\mathbf{x} = 1$ discriminate the other type of agents. In the simulation runs, I assume that the baseline transmission probability is 0.6, the γ_1 value is 6, while the γ_2 value equals to

0.1. This means that the potential pairwise transmission probabilities conditionally on the participants' feature values are as follows.

$$P_{i,j} = \begin{cases} 0.6 \cdot \exp(0), & \text{if } \mathbf{x}_i = 1 \text{ and } \mathbf{x}_j = 1 \\ 0.6 \cdot \exp(0), & \text{if } \mathbf{x}_i = 0 \text{ and } \mathbf{x}_j = 0 \\ 0.6 \cdot \exp(-0.1), & \text{if } \mathbf{x}_i = 0 \text{ and } \mathbf{x}_j = 1 \\ 0.6 \cdot \exp(-6), & \text{if } \mathbf{x}_i = 1 \text{ and } \mathbf{x}_j = 0 \end{cases}$$

With this experimental set up, I simulated 500 similarity based diffusion processes with a random choice of the initial seeder. The random choice of the seeder means that with a 0.3 probability the initial seeder is an agent who discriminates, and with a probability of 0.7, the agent is not discriminating based on the feature value. Because of this, the time dependent expected ratio of infected nodes will be different in those simulation runs when the initial seeder is from the discriminating breed. Simulation results regarding a non homophilous feature arrangement (when Moran's I is roughly 0) is plotted in Figure 8.8, where the discrete points are smoothed out. In the simulation runs, when the initial seeder was discriminating the time dependent expected ratio of infected nodes is lower, then the runs when the initial seeder was a non discriminating agent. In Figure 8.8 this is represented by that the red curve is above the blue one in all of the time periods. It is also observable that only the red curve has sigmoid shape, the blue one (when the seeder is a discriminator) converges to a fully infected state with a much slower pace. The R scripts used for the simulation are attached in Appendix E as Scripts E.28 and E.29.

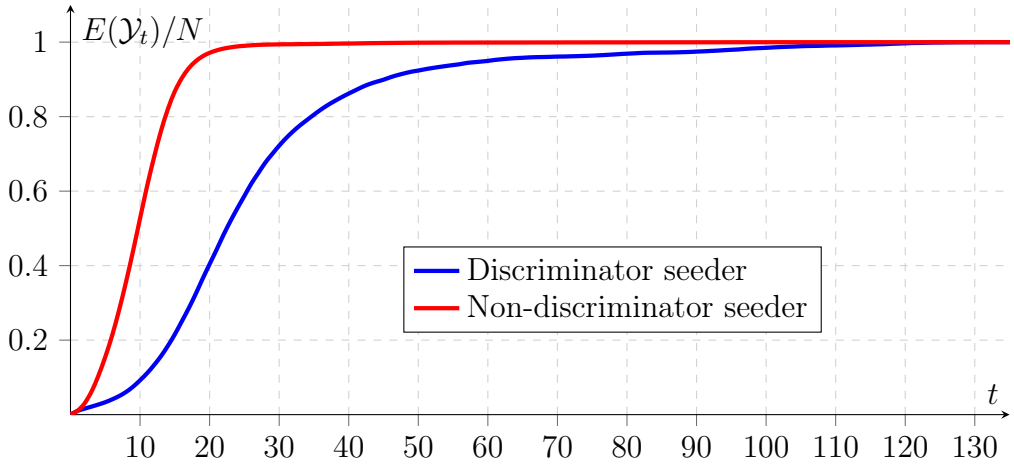


Figure 8.8: The ratio of infected nodes as a function of time – non-homophilous state with discrimination

This difference in the time dependent expected ratio of infected nodes (between discriminator and non-discriminator initiated diffusion) is more remarkable when the network shows homophily. The curves in Figure 8.9 were generated by using the similarity based diffusion model on a homophilous network. The only difference from the previous simulation is that the network shows mild homophily – in each simulation run I rearrange the generic vertex feature in a way that the network shows homophily. The target value of the homophily rearrangement is a Moran's I equal to 0.5. There are three important phenomena that can be inferred based on the curves in Figure 8.9. First, the difference between the two types of diffusion is stronger. Second, homophily also effect the expected convergence time to a perfectly infected state. Both in case of the discriminating and the non-discriminating seeder initiated diffusion, the expected convergence time is higher than it was when homophily was not present – for comparison, see the curves in Figure 8.8. Third, the sigmoid shape of the time–expected ratio of infected nodes curve disappears if the seeder is a discriminator and the network shows homophily regarding the feature. The R Script that generated the diffusion process and the data used for the plot is enclosed in Appendix E as Scripts E.28 and E.29.

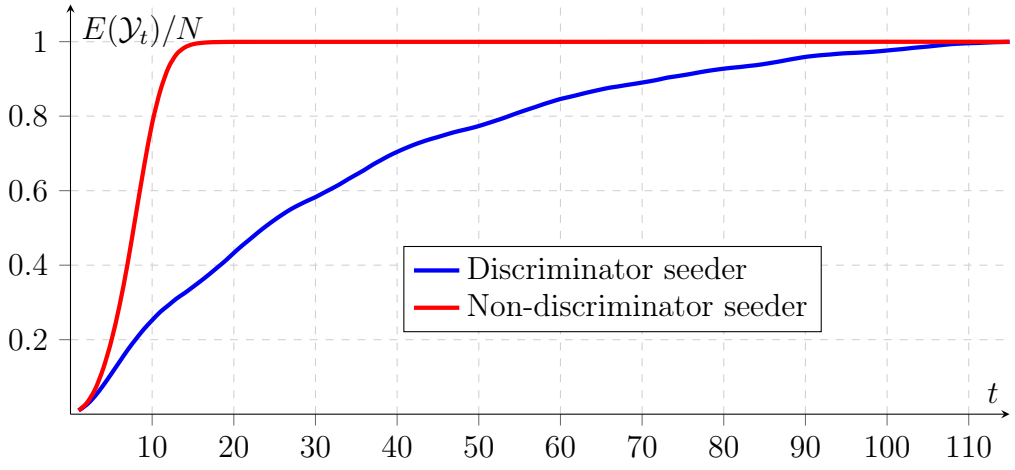


Figure 8.9: The ratio of infected nodes as a function of time – homophilous state with discrimination

Chapter 9: Conclusion

This chapter gives an overview of my thesis regarding the main results, the possible policy relevant applications of my ideas, the limitations and plausible extensions of my research. The main findings and the novel achievements of the thesis are summarized in Section 9.1. The policy relevant implications of the thesis, possible applications of the proposed algorithms and the diffusion model are the subject of Section 9.2. The major theoretical and empirical limitations of my research are highlighted in Section 9.3. The chapter concludes in Section 9.4 with the discussion of possible improvements of the algorithms, models and experiments done in my thesis.

9.1 Summary of findings

The thesis in hand had three closely related research questions that it wanted to investigate. The first one was about algorithms and the properties of these algorithms that can rearrange the sole generic vertex feature of networks in a way that the resulting feature shows homophily while the feature's distribution and the network topology are unchanged. My thesis coined these algorithms as homophily rearrangement algorithms. The second research question was about the multivariate extension of the algorithms proposed by the first research question and also about the characteristics of these multivariate homophily rearrangement algorithms. The third question dealt with an extension of the susceptible-infected model in a setting where the pairwise transmission probabilities between vertices depended on the generic vertex similarity of the vertices.

The results regarding the first and second research question can be summarized as follows. In Chapter 4 I proposed three new algorithms that can rearrange a single generic vertex features in a way that the network shows homophily regarding them. These algorithms are able to tackle the homophilous rearrangement of binary, categorical, ordinal, count and continuous features on a network which has an arbitrary topology with high relia-

bility. These algorithms were augmented in Chapter 5 to solve multivariate homophily rearrangement problems. Sensitivity analysis of the proposed univariate and multivariate homophily rearrangement algorithms in Chapter 7 established how macro-level parameters effect the results of the homophily rearrangement process. The most important simulation results are enumerated below.

1. The expected and median time (number of iterative steps) needed for univariate homophily rearrangement increases in the absolute value of the target homophily and the size of the network. This is even true when the network has multiple generic vertex features.

2. The feature vectors that result from the univariate homophily rearrangement are practically uncorrelated with each other and the original generic vertex feature.

3. The multivariate homophily rearrangements show that the correlation of the generic vertex features decreases the number of iterative steps needed for solving the homophily rearrangement problem. This holds both for positive and negative generic vertex feature correlation.

4. The results obtained with multivariate homophily rearrangements show that the correlation of the generic vertex features mitigates homophily inertia – the phenomenon that the homophily levels diverge from each other.

To answer the third proposed research question, I defined the the similarity based diffusion model in Chapter 6. In this agent based model, the pairwise transmission probability of the *infection* depended negatively on the dissimilarity of agents' generic vertex features. The model was used for simulations and the simulation results presented in Chapter 8 supported that in this similarity based diffusion setting, homophily propagates the spreading on the network, while heterophily slows down the spreading of the infection. In addition, I proposed a modified version of the similarity based diffusion model in Chapter 6 where certain agents weighted the dissimilarity of other agents up. The simulation results in Chapter 8 obtained with the modified model indicated that homophily slowed down the spreading of the infection if it originated from the discriminating group.

9.2 Policy relevant implications

While the topic of my thesis is theoretical and computational, the introduced algorithms and empirical results have direct applications and also important policy related implications. First I will touch upon the application of the homophily rearrangement simulations.

Second, I will highlight the policy relevant implications of the simulation results obtained by the similarity based diffusion model.

Let us imagine that one wants to investigate with a randomized experiment how homophily changes peer effects and different outcomes in work relationships. The experimental setup includes that participants have to co-operate in a predefined way, the co-working relationships are described by a network, where nodes represent the participants and the existence of co-working is described by the edges. If one separates the participants randomly into two groups, we can assume that the participants in the two groups have traits that are distributed identically in the two groups. These two groups can form two artificially generated network mentioned above in a way that people are assigned to work with others in a preset way. In this case, one would expect that the homophily regarding the generic vertex features (the traits of participants) will be basically the same on the two networks. Nevertheless, one can assign the participants to take certain co-working roles based on the homophily rearrangement of the original assignment.

In line with the above mentioned one can construct a network of co-operations that show homophily or heterophily regarding a single dimension of participants (gender, race or ability among many others) or multiple dimensions. Moreover, if one is able to do a multiple homophily rearrangement in a way that only homophily regarding a single feature is changed, then the effect of other observed traits that are correlated with the feature of interest can be filtered out. However, it should be emphasized that the change in homophily regarding unobserved characteristics cannot be assessed. This method might quantify the effectiveness of firm-level policies about co-working or the effect of peer-based interventions in education. Generally, it is useful in every situation when one wants to control peer-effects. With the application of homophily rearrangement algorithms, the above mentioned experiments can be done on networks of arbitrary size, so experiments such as the one done by Centola (2011) are testable on large networks.

The similarity based diffusion model has no direct applications, but it has important policy implications about the spreading of ideas and innovation on networks that show homophily. Simulation results that were obtained with the similarity based diffusion model show that the spreading of information is way slower when the network is homophilous regarding a certain feature and the initial seeder is in a groups which discriminates based on the feature. An obvious example for this can be a social network of everyday interac-

tions, where one observes race based homophily and discrimination based on race at the same time. The importance of the slower diffusion lies in the fact that the innovation or information might have a life-cycle which is shorter than the the expected convergence time. One example is the spreading of information about new positions in the labor market. If the spreading of the information is sufficiently slow, the position might be already filled when a discriminated person gets to know that the opening existed. Another example can be the adaptation of a technological innovation such as a software or a gadget. By the time the discriminated agents adapt it is possibly seriously outdated. This lag in the adaptation leads to another inefficiency. However, these inefficiencies are bridgeable if the initial seeding of the information happens in multiple communities of the network that do not share the same feature value. In the labor market case it means that the position should be advertised in communities that are different from each other. While in the case of technological innovation, it simply means that early adapters should be in different communities regarding the feature of interest (race, gender, religion or education level).

These above described policy ideas about the connection of homophily, diffusion and possible discrimination could solve certain actual labor market and corporate governance problems. As the findings of Petersen et al. (2000) support it, the labor market referrals are biased towards shared race and gender. To help the diffusion of information about job postings companies might give extra incentives to support successful cross gender and race talent referrals within the company. This way the diffusion process is probably speeded up and potential able applicants hear about the positions in time. Another application of the results obtained with the similarity based diffusion model is related to the findings of Edling et al. (2012) about Scandinavian corporate governance networks. There are two important facts highlighted by Edling et al. (2012). First, boards have homophilous connections to other boards. Second, that the ratio of females increases in boards through diffusion, boards that have connections to other boards with female members are likelier to replace aged out male members with females. This gives place for a reasonable regulation of the gender ratio in the boards.[1] We know that the networks show homophily regarding board composition and assuming similarity based diffusion is reasonable in this case. Because of the before mentioned, the board composition regulation should primarily target those firms that are well connected with others in boards and have traits (e.g. board composition, industry or firm size) that make them extremely similar to their neighbors.

[1]Which is an actual policy target of Scandinavian governments.

9.3 Limitations of the research

There are certain considerable computational and theoretical limitations of the homophily rearrangement algorithms introduced in my thesis. An important shortcoming of the proposed homophily rearrangement algorithms is that both the univariate and multivariate homophily rearrangement algorithms assume that the homophily measurement function needs the adjacency matrix in order to quantify the level of homophily. This slows down the homophily rearrangement algorithms. Such measures of homophily are inefficient, because one has to calculate computationally demanding inner products. Another imperial weakness of the homophily rearrangement algorithms is that the multivariate ones can only have target values that are either all positive or negative. This can be problematic because there are certain networks that show homophily in one dimension while in another one heterophily can be present. An example, is the network of sexual relationships, where relationships are homophilous regarding race and heterophilous with respect to gender.

The empirical investigation of the homophily rearrangement algorithms and the similarity based diffusion model is also limited. These limitations can be summarized in four major points. First, the sensitivity analyses regarding both the homophily rearrangement algorithms and the similarity based diffusion model are all implemented on square lattice. This choice of topology might affect the sensitivity analysis results presented in my thesis. Second, the generic vertex features of interest have Bernoulli and standard normal distributions in my thesis, which is a simplistic assumption about the nature of generic vertex features. Third, the homophily regarding the generic vertex features is measured just by two specific homophily measurement functions – the inbreeding homophily index and Moran's I. The stability of results should be assessed by applying other homophily measurement functions. Fourth, the similarity based diffusion is only analyzed in systems where the nodes have a single generic vertex feature. Correlation of the generic vertex features with each other might influence the diffusion process.

9.4 Further research possibilities

The algorithms and the diffusion model proposed in my thesis are fairly modular – certain elements of them can be changed and the change results in a different algorithm or model. This flexibility allows for extensions of the proposed algorithms in multiple ways. Current results of the empirical investigation about the sensitivity of the homophily rearrangement algorithms and the similarity based diffusion model might be also enriched. There are three main ways that my research can be augmented or improved to hedge its current shortcomings.

First, the proposed algorithms use homophily measurement functions that calculate the level of homophily based on the generic vertex feature and the adjacency matrix of the network. This is a computationally inefficient way of quantifying homophily on the network. Implementing the proposed univariate and multivariate algorithms with homophily measurement functions that quantify homophily based on the generic vertex feature and the edge list could be a novel contribution to my research. Such improvement of the algorithms would allow for the simulation of large-scale homophily rearrangements. This idea would fit with the *Apache Spark GraphX* environment which is a scalable big data analysis tool for network analytics.[2]

Second, the results about the expected convergence time of the similarity based diffusion model and the dispersion of the observed convergence times implies that choosing certain seeders can speed up the similarity based diffusion process. Creating vertex selection heuristics that can choose vertices that initiate similarity based diffusion processes which have short expected convergence times would be an interesting extension of my results. The unique value of these heuristic would lie in the fact that such heuristics might essentially find a fairly optimal way to initiate fast diffusion of the networks. This would be helpful, when one can assume that the diffusion is similarity based and the subject of the spreading is either a technical innovation or an important information with short life-cycle.

Third, as highlighted previously, the sensitivity analyses were mainly implemented for networks that have a simple square lattice topology. Both the similarity based diffusion model and the homophily rearrangement algorithms have properties that were only

[2]For details see `http://spark.apache.org/graphx/`.

demonstrated on lattices. Because of this, my study should be repeated on networks that have topology different from a square lattice in order to prove that the sensitivity analysis results hold more generally. The particular empirical investigation might focus on real world networks or on artificial ones which have topology described by canonical models such as the Erdős-Rényi graph.

Bibliography

J. Balthrop, et al. (2004). 'Technological Networks and the Spread of Computer Viruses'. *Science* **304**(5670):527–529.

A.-L. Barabási (2016). *Network Science*. Cambridge University Press.

A.-L. Barabási & R. Albert (1999). 'Emergence of Scaling in Random Networks'. *Science* **286**(5439):509–512.

A.-L. Barabási & R. Albert (2002). 'Statistical mechanics of complex networks'. *Reviews of Modern Physics* **74**:47–49.

F. Bass (1969). 'A New Product Growth for Model Consumer Durables'. *Management Science* **15**(5):215–227.

P. S. Bearman, et al. (2004). 'Chains of Affection: The Structure of Adolescent Romantic and Sexual Networks'. *American Journal of Sociology* **110**(1):44–91.

G. Bianconi & A.-L. Barabasi (2001). 'Competition and Multiscaling in Evolving Networks'. *Europhysics Letters* **54**(4):436–442.

H. Bisgin, et al. (2010). 'Investigating Homophily in Online Social Networks'. In IEEE (ed.), *Web Intelligence and Intelligent Agent Technology (WI-IAT), IEEE/WIC/ACM International Conference*, vol. 1.

A. Calvo-Armengol & M. O. Jackson (2004). 'The Effects of Social Networks on Employment and Inequality'. *American Economic Review* pp. 426–454.

D. Centola (2011). 'An Experimental Study of Homophily in the Adoption of Health Behavior'. *Science* **334**:1269–1273.

D. Centola & A. van de Rijt (2015). 'Choosing Your network: Social Preferences in an Online Health Community'. *Social Science and Medicine* **125**:19–31.

J. Coleman (1958). 'Relational Analysis: The Study of Social Organizations with Survey Methods'. *Human Organization* **17**:28–36.

T. H. Cormen, et al. (2009). *Introduction to Algorithms*. The MIT Press, Cambridge, Massachusetts.

S. Currarini, et al. (2009). 'An Economic Model of Friendship: Homophily, Minorities, and Segregation'. *Econometrica* **77**(4):1003–1045.

M. G. de Matos, et al. (2014). 'Peer Influence in the Diffusion of iPhone 3G over a Large Social Network'. *Management Information Systems Quarterly* **4**:1103–1133.

M. M. Deza & E. Deza (2009). *Encyclopedia of Distances*. Springer.

D. Easley & J. Kleinberg (2010). *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*. Cambridge University Press.

F. Echenique & R. G. Fryer (2007). 'A Measure of Segregation Based on Social Interactions.'. *Quarterly Journal of Economics* **122**(2):441–485.

C. Edling, et al. (2012). *The Small Worlds of Corporate Governance*, chap. Testing the Old Boys Network: Diversity and Board Interlocks in Scandinavia, pp. 183–202. MIT Press.

J. Epstein (1986). *Process and Outcome in Peer Relationships*, chap. Friendship Selection: Developmental and Environmental Influences., pp. 129–160. New York: Academic Press.

P. Erdős & A. Rényi (1960). 'On the Evolution of Random Graphs'. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences* **5**:17–61.

G. Fagiolo, et al. (2007). 'Segregation in Networks'. *Journal of Economic Behavior and Organization* **64**:316–336.

R. M. Fernandez & I. Fernandez-Mateo (2006). 'Networks, Race, and Hiring'. *American Sociological Review* **71**(1):42–71.

L. Freeman (1972). 'Segregation in Social Networks'. *Sociological Methods and Research* **6**(411-427).

L. Freeman (1978). 'On Measuring Systematic Integration'. *Connections* **12**(1):13–14.

C. Freshtman & U. Gneezy (2001). 'Discrimination in a Segmented Society: an Experimental Approach'. *Quartely Journal of Economics* pp. 351–377.

B. Frey & D. Dueck (2007). 'Clustering by Passing Messages Between Data Points'. *Science* **315**(5814):972–976.

R. C. Geary (1954). 'The Contiguity Ratio and Statistical Mapping'. *The Incorporated Statistician* **5**(3):115–145.

B. Golub & M. O. Jackson (2012). 'Network Structure and the Speed of Learning'. *Annals of Economics and Statistics* (107-108):33–48.

M. C. Gonzalez, et al. (2007). 'Community Structure and Ethnic Preferences in School Friendship Networks'. *Physica A: Statistical Mechanics and its Applications* **379**(1):307–316.

J. Gower (1971). 'A General Coefficient of Similarity and Some of It's Properties'. *Biometrics* **27**:857–872.

M. Granovetter (1978). 'Threshold Models of Collective Behavior'. *American Journal of Sociology* pp. 1420–1443.

S. Gupta, et al. (1989). 'Networks of Sexual Contacts: Implications for the Pattern of Spread of HIV'. *AIDS* **3**(12):807–818.

Y. Halberstam & B. Knight (2014). 'Homophily, Group Size, and the Diffusion of Political Information in Social Networks'. *National Bureau of Economic Research* .

K. M. Harris (2009). 'The National Longitudinal Study of Adolescent to Adult Health (Add Health), Waves I and II, 1994–1996; Wave III, 2001–2002; Wave IV, 2007-2009'. Chapel Hill, NC: Carolina Population Center, University of North Carolina at Chapel Hill. DOI: 10.3886/ICPSR27021.v9.

S. Holzhauer, et al. (2013). 'Considering Baseline Homophily When Generating Spatial Social Networks for Agent-Based Modelling'. *Computational and Mathematical Organization Theory* **19**(2):128–150.

M. O. Jackson (2010). *Social and Economic Networks*. Princeton University Press.

M. O. Jackson, et al. (2016). 'The Economic Consequences of Social Network Structure'. Available at SSRN: http://ssrn.com/abstract=2467812.

M. Kalmijn & H. Flap (2001). 'Assortative Meeting and Mating: Unintended Consequences of Organized Settings for Partner Choices'. *Social Forces* **79**(4):1289–1312.

A. Kamath & R. Cowan (2015). 'Social Cohesion and Knowledge Diffusion: Understanding the Embeddedness–Homophily Association'. *Socio-Economic Review* **13**(4):723–746.

G. Kao & K. Joyner (2004). 'Do Race and Ethnicity Matter among Friends? Activities among Interracial, Interethnic, and Intraethnic Adolescent Friends.'. *Sociological Quarterly* **45**(3):557–573.

D. Kempe, et al. (2003). 'Maximizing the Spread of Influence Through a Social Network'. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining.*, pp. 137–146. ACM.

C. Kenyon & R. Colebunders (2013). 'Birds of a Feather: Homophily and Sexual Network Structure in Sub-Saharan Africa'. *Internation Journal of Sexually Transmitted Diseases and AIDS* **24**(3):211–215.

B. Kogut, et al. (2012). *The Small Worlds of Corporate Governance*, chap. Generating Rules and the Social Science of Governance, pp. 259–299. MIT Press.

D. Krackhardt & R. N. Stern (1988). 'Informal Networks and Organizational Crises: An Experimental Simulation'. *Social Psychology Quarterly* **51**(2):123–140.

E. Laumann & Y. Youm (1999). 'Racial and Ethnic Group Differences in the Prevalence of Sexually Transmitted Diseases in the United States: A Network Explanation'. *Sexually Transmitted Diseases* **26**(5):250–261.

P. F. Lazarsfeld & R. K. Merton (1954). 'Friendship as a Social Process: A Substantive and Methodological Analysis'. *Freedom and Control in Modern Society* **18**(1):18–66.

R. D. Mare (1991). 'Five decades of educational assortative mating'. *American sociological review* **15-32**.

A. Mayer & S. L. Puller (2008). 'The Old Boy (and Girl) Network: Social Network Formation on University Campuses'. *Journal of Public Economics* **92**:329–347.

M. McPherson, et al. (2001). 'Birds of a Feather: Homophily in Social Networks'. *Annual Review of Sociology* **27**:415–444.

J. Moody (2001). 'Peer Influence Groups: Identifying Dense Clusters in Large Networks'. *Social Networks* **23**(261-283).

P. A. P. Moran (1950). 'Notes on Continuous Stochastic Phenomena'. *Biometrika* **37**(1):17–23.

S. Navlakha & C. Kingsford (2010). 'The Power of Protein Interaction Networks for Associating Genes with Diseases'. *Bioinformatics* **26**(8):1057–1063.

H. Noel & B. Nyhan (2011). 'The Unfriending Problem: The Consequences of Homophily in Friendship Retention for Causal Estimates of Social Influence'. *Social Networks* **311**:211–218.

R. Noldus & P. V. Mieghem (2015). 'Assortativity in Complex Networks'. *Journal of Complex Networks* **3**(7):507–542.

H. W. Park, et al. (2001). 'Political Communication Structure in Internet Networks - A Korean Case'. *Sungkok Journalism Review* **11**:67–89.

H. W. Park & M. Thelwall (2003). 'Hyperlink Analyses of the World Wide Web: A Review'. *Journal of Computer-Mediated Communication* **8**(4).

T. Petersen, et al. (2000). 'Offering a Job: Meritocracy and Social Networks'. *American Journal of Sociology* **106**(3):763–816.

A. P. Quayle, et al. (2006). 'Modeling Network Growth with Assortative Mixing'. *The European Physical Journal B - Condensed Matter and Complex Systems* **50**(4):617–630.

H. Rahmani, et al. (2011). *Advances in Intelligent Data Analysis X*, chap. Collaboration-based Function Prediction in Protein-Protein Interaction Networks, pp. 318–327. Springer.

C. Roth & J.-P. Cointet (2010). 'Social and Semantic Coevolution in Knowledge Networks'. *Social Networks* **32**(1):16–29.

T. Schelling (1969). 'Models of Segregation'. *American Economic Review* **59**:488–493.

W. Shrum, et al. (1988). 'Friendship in School: Gender and Racial Homophily'. *Sociology of Education* **61**(4):227–239.

B. Silverman (1986). *Monographs on Statistics and Applied Probability*, chap. Density Estimation for Statistics and Data Analysis, pp. 1–22. London: Chapman and Hal.

P. van Eck & W. Jager (2010). 'Social Network Structures in Agent Based Modelling: Finding an Optimal Structure Based on Survey Data (or Finding the Network That Does Not Exist).'. In *Proceedings of the 3rd World Congress on social simulation*.

S. G. Vandenberg (1972). 'Assortative Mating, or Who Marries Whom?'. *Behavior Genetics* **2**(2-3):127–157.

D. Watson, et al. (2004). 'Match Makers and Deal Breakers: Analyses of Assortative Mating in Newlywed Couples.'. *Journal of Personality* **72**(5):1029–1068.

D. J. Watts (2002). 'A Simple Model of Global Cascades on Random Networks'. *Proceedings of the National Academy of Sciences of the United States of America* **99**(9):5766–5771.

A. Wimmer & K. Lewis (2010). 'Beyond and Below Racial Homophily: Exponential Random Graphs Models of a Friendship Network Documented on Facebook'. *American Journal of Sociology* **116**(2):583–642.

M. Yavas & G. Yusel (2014). 'Impact of Homophily on Diffusion Dynamics Over Social Networks'. *Social Science Computer Review* **33**(3):354–372.

Appendix A: Notations

V – Set of vertices in a network

E – Set of edges in a network

$\mathbf{x}$ – Generic vertex feature vector

$\mathbf{x}_i$ – The i^{th} element of the generic feature vector $\mathbf{x}$

$\mathbf{X}$ – Matrix of generic vertex features

$\mathbf{X}_i$ – The i^{th} row of generic feature matrix $\mathbf{X}$

$\widetilde{\mathbf{x}}$ – Temporary generic vertex feature

$\widetilde{\mathbf{X}}$ – Temporary feature matrix

$\mathrm{G}(V, E, \mathbf{x})$ – Network with a single generic feature

$\mathrm{G}(V, E, \mathbf{X})$ – Network with a feature matrix

$\mathbf{W}$ – Adjacency matrix of the network

$\mathrm{G}(\mathbf{W}, \mathbf{x})$ – Network with a single generic feature

$\mathrm{G}(\mathbf{W}, \mathbf{X})$ – Network with a feature matrix

N – Number of elements in feature vector $\mathbf{x}$

t – Step of the homophily rearrangement/diffusion algorithm

c – Level of homophily

$\mathcal{C}$ – Categorical homophily levels (criteria)

$\mathcal{H}$ – Homophily measurement function

$\mathcal{H}_U$ – Universal homophily measurement function

$\mathcal{H}_C$ – Categorical homophily measurement function

ω_t – Level of homophily at step t.

Ω_t – Level of categorical homophily values at step t.

ϕ – Universal target homophily value

Φ – Categorical target homophily vector

$(\mathcal{C}_1, \ldots, \mathcal{C}_p)$ – Homophily profile

$\mathcal{E}(\mathcal{H}_1(\mathbf{x}_1, \mathbf{W}), \ldots, \mathcal{H}_p(\mathbf{x}_p, \mathbf{W}))$ – Ensemble homophily measurement function

$(\Phi_1, \ldots, \Phi_p)$ – Homophily target profile

$\mathcal{S}_U(\phi, \mathbf{x}, \mathbf{W})$ – Universal vertex pair selection heuristic

$\mathcal{S}_C(\Phi, \mathbf{x}, \mathbf{W})$ – Categorical vertex pair selection heuristic

$\mathcal{X}^{\uparrow}$ – Above the mean control value

$\mathcal{X}^{\downarrow}$ – Below the mean control value

$\mathcal{X}^{m}$ – Within group control value

$\mathcal{X}^{\not m}$ – Outside the group control value

P_0 – Baseline transmission probability

$P_{i,j}$ – Pairwise transmission probability from agent i to j

$d(\mathbf{X}_i, \mathbf{X}_j)$ – Dissimilarity of agents i and j

γ – Transmission sensitivity to dissimilarity

Ψ – Base function

$\mathcal{K}$ – Set of newly infected nodes

$\mathcal{I}$ – Set of newly infected nodes by node i

$\mathbf{Y}_t$ – Infestion state vector at time period t

$\mathbf{Y}_{t,i}$ – Infection state of node i at time period t

$\mathcal{Y}_t$ – Number of infected nodes at time period t

$\mathrm{G}_i^*(\mathbf{X}^*, \mathbf{W}^*)$ – Vertex i induced star network

$\mathcal{N}_{\mathcal{G}}$ – First order neighborhood function

d_i^* – Average feature dissimilarity from vertex i

Δ_i – Degree corrected feature dissimilarity of vertex i

Θ – Stopping flag

Appendix B: Auxiliary algorithms

Data: Homophily target value ϕ, the feature value $\mathbf{x}$ and the weighted adjacency matrix $\mathbf{W}$.

Result: Pair of indices i and j participating in the feature value rearrangement.

$N \Leftarrow |\mathbf{x}|$
$V \Leftarrow \{1, \ldots, N\}$
if $\phi > 0$ **then**
    $\mathcal{X}^{\uparrow} \Leftarrow -\infty$
    $\mathcal{X}^{\downarrow} \Leftarrow -\infty$
else
    $\mathcal{X}^{\uparrow} \Leftarrow \infty$
    $\mathcal{X}^{\downarrow} \Leftarrow \infty$
end
for v **in** V **do**
    if $\mathbf{x}_v \geq \overline{\mathbf{x}}$ **then**
        if $[(\phi > 0$ **and** $\deg(v) \cdot d_v^* > \mathcal{X}^{\uparrow})$ **or** $(\phi < 0$ **and** $\deg(v) \cdot d_v^* < \mathcal{X}^{\uparrow})]$ **then**
            $i \Leftarrow v$
            $\mathcal{X}^{\uparrow} \Leftarrow \deg(v) \cdot d_v^*$
        end
    else
        if $[(\phi > 0$ **and** $\deg(v) \cdot d_v^* > \mathcal{X}^{\downarrow})$ **or** $(\phi < 0$ **and** $\deg(v) \cdot d_v^* < \mathcal{X}^{\downarrow})]$ **then**
            $j \Leftarrow v$
            $\mathcal{X}^{\downarrow} \Leftarrow \deg(v) \cdot d_v^*$
        end
    end
end

Algorithm 11: Pseudo code of the universal vertex selection algorithm

Data: Homophily target vector Φ, the feature value $\mathbf{x}$ and the weighted adjacency matrix $\mathbf{W}$.

Result: Pair of indices i and j participating in the feature value rearrangement.

$N \Leftarrow |\mathbf{x}|$
$V \Leftarrow \{1, \ldots, N\}$
$\mathcal{M} \Leftarrow \{1, \ldots, M\}$
$m \Leftarrow U(\mathcal{M})$
if $\Phi \succ \mathbf{0}$ **then**
    $\mathcal{X}^m \Leftarrow -\infty$
    $\mathcal{X}^{\not m} \Leftarrow -\infty$
else
    $\mathcal{X}^m \Leftarrow \infty$
    $\mathcal{X}^{\not m} \Leftarrow \infty$
end
for v **in** V **do**
    if $\mathbf{x}_v = m$ **then**
        if $[(\Phi \succ \mathbf{0}$ **and** $\deg(v) \cdot d_v^* > \mathcal{X}^m)$ **or** $(\Phi \prec \mathbf{0}$ **and** $\deg(v) \cdot d_v^* < \mathcal{X}^m)]$ **then**
            $i \Leftarrow v$
            $\mathcal{X}^m \Leftarrow \deg(v) \cdot d_v^*$
        end
    else
        if $[(\Phi \succ \mathbf{0}$ **and** $\deg(v) \cdot d_v^* > \mathcal{X}^{\not m})$ **or** $(\Phi \prec \mathbf{0}$ **and** $\deg(v) \cdot d_v^* < \mathcal{X}^{\not m})]$ **then**
            $j \Leftarrow v$
            $\mathcal{X}^{\not m} \Leftarrow \deg(v) \cdot d_v^*$
        end
    end
end

Algorithm 12: Pseudo code of the categorical vertex selection algorithm

Appendix C: Tables

Table C.1: Classification of homophily measurement functions

Measure's name	**Type**	**Normalized**	**Additional information**
Moran's I	Universal	Yes	Moran (1950)
Transformed Geary's C	Universal	Yes	Geary (1954)
End-node correlations	Universal	Yes	Noldus & Mieghem (2015)
External-internal links index	Categorical	Yes	Krackhardt & Stern (1988)
Homophily index	Categorical	Yes	Coleman (1958)
Segregation matrix index	Categorical	Yes	Freshtman & Gneezy (2001)
Gupta-Anderson-May index	Categorical	No	Gupta et al. (1989)
Segregation index	Universal/Categorical	No	Freeman (1978)
Homophily test	Categorical	No	Easley & Kleinberg (2010)
Inbreeding homophily index	Categorical	Yes	Currarini et al. (2009)
Spectral segregation index	Individual	No	Echenique & Fryer (2007)

Appendix D: Figures

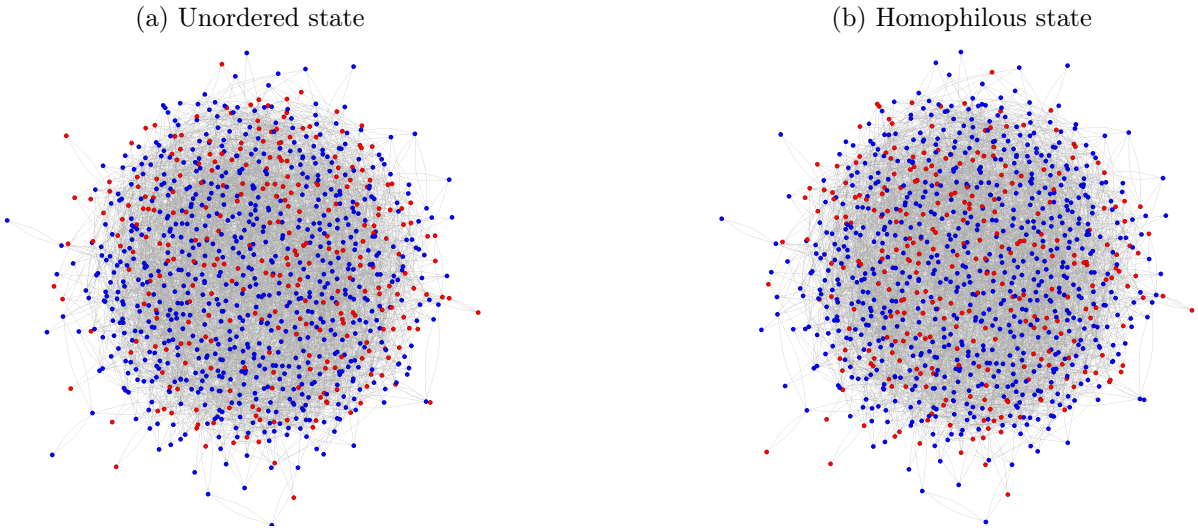


Figure D.1: Heuristic homophily rearrangement of a binary feature – ER topology

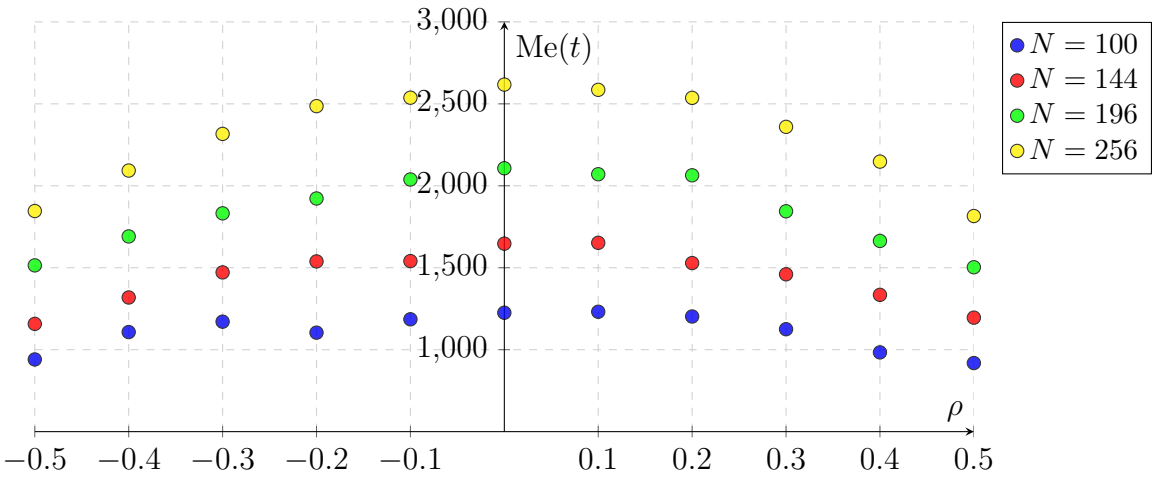


Figure D.2: Median solution time of the multivariate heuristic homophily rearrangement algorithms as a function of feature correlation and lattice size

Appendix E: R Scripts

E.1 Heuristic algorithms

```
randomkick_generator ← function(Phi,X,W){
NX ← length(X)
tindex ← 0
Cal_C ← Moran(X,W)[1]
while ((Phi>Cal_C & Phi>0)|(Phi<Cal_C & Phi< 0)){
    Tilde_X ← X
    tindex ← tindex + 1
    i ← sample(1:NX,1)[1]
    j ← sample(1:NX,1)[1]

    if (Tilde_X[i] != Tilde_X[j]){
        Tilde_X[i] ← X[j]
         Tilde_X[j] ← X[i]
         Tilde_Cal_C ← Moran(Tilde_X,W)[1]

         if ((Tilde_Cal_C>Cal_C & Phi>0)|(Tilde_Cal_C<Cal_C & Phi<0)){
              print(Tilde_Cal_C)
              X[i] ← Tilde_X[i]
              X[j] ← Tilde_X[j]
              Cal_C ← Tilde_Cal_C
          }
     }
}
return(X)}
```

Script E.1: R implementation of the heuristic rearrangement algorithm for a general homophily measure function

```
randomkick_generator ← function(Phi, X, W){
  NX ← length(X)
  t ← 0
  Cal_C ← HI_measure(W, Tilde_X)
  Z ← rep(0,length(Cal_C))
  while ( (any(Phi > Cal_C)& all(Phi >Z))|(any(Phi < Cal_C)& all(Phi
      < Z)) ){
    Tilde_X ← X
    t ← t + 1
    i ← sample(1:NX,1)[1]
    j ← sample(1:NX,1)[1]
    if (Tilde_X[i] != Tilde_X[j]){
      Tilde_X[i] ← X[j]
      Tilde_X[j] ← X[i]
      Tilde_Cal_C ← HI_measure(W, Tilde_X)
      if ((all(Tilde_Cal_C > Cal_C) & all(Phi > Z))|(all(Tilde_Cal_C
          < Cal_C) & all(Phi < Z)) ){
        X[i]←Tilde_X[i]
        X[j]←Tilde_X[j]
        Cal_C←Tilde_Cal_C
      }
    }
  }
  return(X)}
```

Script E.2: R implementation of the heuristic rearrangement algorithm for a categorical homophily measure function

E.2 Heuristic algorithms with bag of indices

```
randomkick_generator ← function(Phi, X, network){
  Bag ←1:length(X)
  time ← 0
  Cal_C ← Moran(X,W)[1]
  Z ← rep(0,length(Cal_C))
  while ((Phi>Cal_C & Phi>0)|(Phi<Cal_C & Phi< 0)){
    Tilde_X ← X
    time ← time + 1
    i ← sample(Bag,1)[1]
    j ← sample(Bag,1)[1]
    if (Tilde_X[i] != Tilde_X[j]){
      Tilde_X[i] ← X[j]
      Tilde_X[j] ← X[i]
      Tilde_Cal_C ← Moran(Tilde_X,W)[1]
      if ((Tilde_Cal_C>Cal_C&Phi>0)|(Tilde_Cal_C<Cal_C&Phi<0)){
        X[i] ← Tilde_X[i]
        X[j] ← Tilde_X[j]
        Cal_C ← Tilde_Cal_C
        Bag ← Bag[Bag!=i]
        Bag ← Bag[Bag!=j]
      }
    }
  }
return(X)}
```

Script E.3: R implementation of the heuristic rearrangement algorithm with bag of indices for a general homophily measure function

```
randomkick_generator ← function(Phi, X, network){
  Bag ←1:length(X)
  t ← 0
  Cal_C ← HI_measure(network,X)
  Z ← rep(0,length(Cal_C))
  while ((any(Phi>Cal_C)&all(Phi>Z))|(any(Phi<Cal_C)&all(Phi<Z))){
    Tilde_X ← X
    t ← t + 1
    i ← sample(Bag,1)[1]
    j ← sample(Bag,1)[1]

    if (Tilde_X[i] != Tilde_X[j]){
      Tilde_X[i] ← X[j]
      Tilde_X[j] ← X[i]
      Tilde_Cal_C ← HI_measure(network,Tilde_X)

      if ((all(Tilde_Cal_C>Cal_C)&all(Phi>Z))|(all(Tilde_Cal_C<Cal_C
          )&all(Phi<Z))){
        X[i] ← Tilde_X[i]
        X[j] ← Tilde_X[j]
        Cal_C ← Tilde_Cal_C
        Bag ← Bag[Bag!=i]
        Bag ← Bag[Bag!=j]
      }
    }
  }
return(X)}
```

Script E.4: R implementation of the heuristic rearrangement algorithm with bag of indices for a categorical homophily measure function

E.3 Greedy algorithms

```
greedy_universal ← function(phi,X,W){
Theta ← 0
t ← 0
c ← Moran(X,W)
omega ← c()
while(((phi >= c & phi>0) | (phi <= c & phi < 0)) & (Theta == 0) ){
  t ← t +1
  X_tilde ← X
  i ← selector_universal(phi,X,W)[1]
  j ← selector_universal(phi,X,W)[2]
  X_tilde[i] ← X[j]
  X_tilde[j] ← X[i]
  c_tilde ← Moran(X_tilde,W)
  if ((c_tilde >c & phi > 0) |(c_tilde < c & phi < 0)){
    X[i] ← X_tilde[i]
    X[j] ← X_tilde[j]
    c ← c_tilde
  }else{
    Theta ← 1
  }
  omega[t] ← c
  }
return(X)}
```

Script E.5: R implementation of the heuristic rearrangement algorithm with bag of indices for a general homophily measure function

```
selector_universal ← function(phi,X,W){
  N ← length(X)
  V ← 1:N
   if (phi>0){
  X_above ← -(10^10)
  X_below ← -(10^10)
  } else{
  X_above ← (10^10)
  X_below ← (10^10)
  }
  X_me ← mean(X)
   for (i in 1:N){
     dis ← sum(abs(X[W[,i]==1]-X[i]))
     if(X[i] >= X_me ){
     if((phi > 0 & dis > X_above) | (phi <0 & dis <X_above)){
     out_i ← i
     X_above ← dis
     }
     }else{
       if((phi > 0 & dis > X_below) | (phi <0 & dis <X_below)){
         out_j ← i
         X_below ← dis
       }
     }
  }
return(c(out_i,out_j))
}
```

Script E.6: R implementation of the heuristic rearrangement algorithm with bag of indices for a general homophily measure function

```
greedy_categorical ← function(Phi,X,W){
  network ← graph.adjacency(W)
  Theta ← 0
  t ← 0
  C ← HI_measure(network,X)
  Z ← rep(0,length(C))
  while((any(Phi > C)& all(Phi >Z))|(any(Phi < C)& all(Phi < Z)) & (
      Theta == 0) ){
    t ← t + 1
    print(C)
    X_tilde ← X
    i ← selector_categorical(phi,X,W)[1]
    j ← selector_categorical(phi,X,W)[2]
    X_tilde[i] ← X[j]
    X_tilde[j] ← X[i]
    C_tilde ← HI_measure(network,X_tilde)
    if ((all(C_tilde > C) & all(Phi > Z))|(all(C_tilde < C) & all(
        Phi < Z))){
      X[i] ← X_tilde[i]
      X[j] ← X_tilde[j]
      C ← C_tilde
    }else{
      Theta ← 1
    }
  }
  return(X)}
```

Script E.7: R implementation of the heuristic rearrangement algorithm with bag of indices for a categorical homophily measure function

```
selector_categorical ← function(Phi,X,W){
  N ← length(X)
  V ← 1:N
  M ← 1:length(unique(X))
  m ← sample(unique(X),1)
  Z ← rep(0,length(Phi))
  if (all(Phi>Z)){
    X_m ← -(10^10)
    X_notm ← -(10^10)
  } else{
    X_m ← (10^10)
    X_notm ← (10^10)
  }
  for (i in 1:N){
    dis ← sum(abs(X[W[,i]==1]-X[i]))
    if(X[i] == m){
      if((all(Phi > Z) & (dis > X_m)) | (all(Phi < Z) & (dis <X_m)))
          {
        out_i ← i
        X_m ← dis
      }
    }else{
      if((all(Phi > Z) & (dis > X_notm)) | (all(Phi < Z) & (dis
          <X_notm))){
        out_j ← i
        X_notm← dis
      }
    }
  }
  return(c(out_i,out_j))
}
```

Script E.8: R implementation of the heuristic rearrangement algorithm with bag of indices for a categorical homophily measure function

E.4 Heuristic multivariate algorithm

```
Moran_Vector ← function(X,W){
  Out_vector ←c()
  for (i in 1:ncol(X)){
    Out_vector[i] ←Moran(X[,i],W)
  }
  return(Out_vector)
}
randomkick_generator_multi ← function(Phi,X,W){
  NX ← nrow(X)
  t ← 0
  Cal_C ← Moran_Vector(X,W)
  Z ← rep(0,length(Cal_C))
  while ( (any(Phi > Cal_C)& all(Phi >Z))|(any(Phi < Cal_C)& all(Phi
      < Z)) ){
    Tilde_X ← X
    t ← t + 1
    i ← sample(1:NX,1)[1]
    j ← sample(1:NX,1)[1]
    if (Tilde_X[i,] != Tilde_X[j,]){
      Tilde_X[i,] ← X[j,]
      Tilde_X[j,] ← X[i,]
      Tilde_Cal_C ← Moran_Vector(Tilde_X,W)
      if ((all(Tilde_Cal_C > Cal_C) & all(Phi > Z))|(all(Tilde_Cal_C
          < Cal_C) & all(Phi < Z)) ){
        X[i,] ← Tilde_X[i,]
        X[j,] ← Tilde_X[j,]
        Cal_C ← Tilde_Cal_C
      }
    }
  }
  return(t)}
```

Script E.9: R implementation of the heuristic rearrangement algorithm for a general homophily measure function for multivariate networks

E.5 Heuristic multivariate algorithm with bag of indices

```
randomkick_generator_multi_bag ← function(Phi,X,W){
  Bag ←1:nrow(X)
  NX ← nrow(X)
  t ← 0
  Cal_C ← Moran_Vector(X,W)
  Z ← rep(0,length(Cal_C))
  while ( (any(Phi > Cal_C)& all(Phi >Z))|(any(Phi < Cal_C)& all(Phi
      < Z)) ){
    Tilde_X ← X
    t ← t + 1
    i ← sample(Bag,1)[1]
    j ← sample(Bag,1)[1]

    if (Tilde_X[i,] != Tilde_X[j,]){
      Tilde_X[i,] ← X[j,]
      Tilde_X[j,] ← X[i,]
      Tilde_Cal_C ← Moran_Vector(Tilde_X,W)

      if ((all(Tilde_Cal_C > Cal_C) & all(Phi > Z))|(all(Tilde_Cal_C
          < Cal_C) & all(Phi < Z)) ){
        X[i,] ← Tilde_X[i,]
        X[j,] ← Tilde_X[j,]
        Cal_C ← Tilde_Cal_C
        Bag ← Bag[Bag!=i]
        Bag ← Bag[Bag!=j]
      }
    }
  }
  return(t)}
```

Script E.10: R implementation of the heuristic rearrangement algorithm with bag of indices for a general homophily measurement function for multivariate networks

E.6 Similarity based diffusion model

```
similarity_based_diffusion ← function(g,X,gamma,P_0,ran,non_ran){
  B ← c(1:vcount(g))
  if (ran == TRUE){Btilde ← c(sample(B, 1))}
  else {Btilde ← c(non_ran)}
  B ← setdiff(B, Btilde)
  t ← 0
  N ← vcount(g)
  while (N != length(Btilde) ){
    print(paste0("The time period is: ",t))
    print(paste0("The number of infected nodes: ", length(Btilde)))
    K ← c()
    t ← t + 1
    for (i in Btilde){
      I ← c()
      N_G_I ← neighborhood(g,1,(V(g)[i]))[[1]]
      for(j in N_G_I){
        P ← runif(1,0,1)
        if (ncol(X)==1){Pij ← P_0*exp(gamma[i]*sum(abs(X[i]-X[j])))}
        else{Pij ← P_0*exp(gamma[i]*sum(abs(X[i,]-X[j,])))}
        if(P < Pij){I ← union(I,j)}
      }
      K ← union(K,I)
    }
  B ← setdiff(B,K)
  Btilde ← union(Btilde,K)
  }
}
```

Script E.11: R implementation of the similarity based diffusion model

E.7 Simulations

E.7.1 Univariate homophily rearrangement

```
library(igraph)
set.seed(2016)
g <- erdos.renyi.game(1000, 3000, type="gnm")
W<- as.matrix(get.adjacency(g))
X_1 <- sample(c(0,1), vcount(g), replace = TRUE)
X_temp_1 <- randomkick_generator(0.5, X_1, W)
g_2 <- barabasi.game(1000, 1, 2)
W_2 <- as.matrix(get.adjacency(g_2))
X_2 <- sample(c(0,1), vcount(g_2), replace = TRUE)
X_temp_2 <- randomkick_generator(0.5, X_2, W_2)
```

Script E.12: R implementation of the homophily rearrangements

```
Phi <- 0.5
means <- data.frame(matrix(0,10,30))
medians <- data.frame(matrix(0,10,30))
for ( i in 1:10){
  Dimensions <- 10+i
  W<- Weightgenerator(Dimensions)[[1]]
  for (j in 1:30){
    print(paste0("The number of cells is: ",(10 + i)*(10 + i)))
    print(paste0("The ratio of black cells is: ",0.2+0.01*j))
    Ratio <- 0.2 + 0.01*j
    X <- Valuegenerator(Dimensions,Ratio)[[1]]
    out <- c()
    for (k in 1:200){
      print(k)
      out[k] <- randomkick_generator(Phi,X,W)[1]
    }
    means[i,j] <- mean(out)
    medians[i,j] <- median(out)
  }
}
```

Script E.13: R implementation of the system size – feature entropy experiment

```
dimensions ← 10
means ← c()
medians ← c()
W← Weightgenerator(dimensions)[[1]]
for (i in 1:101){
  Phi ← (-0.51+0.01*i)
  X ← sample(c(0,1),100,replace=TRUE)
  print(paste0("The current run is: ",Phi))
  out_1 ← out_2 ← c()
  for (k in 1:200){
    print(k)
    out_1[k] ← Greedy_Generator_Function(Phi,X,W)
    out_2[k] ← Heuristic_Generator_Function(Phi,X,W)
  }
  means_1[i] ← mean(out_1)
  medians_1[i] ← median(out_1)
  diffs_1[i] ← means_1[i]-medians_1[i]
  means_2[i] ← mean(out_2)
  medians_2[i] ← median(out_2)
  diffs_2[i] ← means_2[i]-medians_2[i]
}
t ← (-50:50)/100
phi_dependence_greedy ← cbind(t, means_1, medians_1, diffs_1)
phi_dependence_heuristic ← cbind(t,means_2,medians_2,diffs_2)
write.csv(phi_dependence_greedy,file="greedy_phi_dep.csv")
write.csv(phi_dependence_heuristic,file="heuristic_phi_dep.csv")
```

Script E.14: R implementation of the target homophily expected and median solution time experiment

```
links ← read.table(paste0("addHealth83.txt"),header=TRUE)
attributes ←  read.table(paste0("addHealth83Attr.txt"),header=TRUE)
colnames(links) ← c("in","out","weight")
colnames(attributes) ← c( "id","sex", "race", "grade")
network ← graph.data.frame(links[,1:2], directed = FALSE)
attributes ← attributes[as.numeric(V(network)$name) ,]
Y ← attributes$sex
Y[Y==0] ← 1
Y ← Y-1
W ← as(get.adjacency(network),"matrix")
out ← c()
for (i in 1:1000){
set.seed(i)
Phi ← 0.5
X ← sample(Y)
print(paste0("The current run is: ",i))
out ← randomkick_generator(Phi,X,W)
write.csv(out,file=paste0("./Heuristic_generations/Same_more",i,"
    .csv"),row.names = FALSE)
}
Concatenated ← data.frame(matrix(0,1268,1000))
for (i in 1:1000){
  Concatenated[,i] ← read.csv(paste0("./Heuristic_generations/
      Same_more",i,".csv"))
}
bet ← betweenness(network)
coren ← coreness(network)
bet_out_cor ← c()
coren_out_cor ← c()
for (i in 1:1000){
  coren_out_cor[i] ← cor(coren,Concatenated[,i])
  bet_out_cor[i] ← cor(bet,Concatenated[,i])
}
Outer ← data.frame(matrix(0,512,8))
Outer[,1] ← density(coren_out_cor)$x
Outer[,2] ← density(coren_out_cor)$y
Outer[,3] ← density(bet_out_cor)$x
```

```
Outer[,4] <- density(bet_out_cor)$y
Outer[,5] <- paste0("(",Outer[,1],",",Outer[,2],")")
Outer[,6] <- paste0("(",Outer[,3],",",Outer[,4],")")
mu_1 <- mean(coren_out_cor)
sigma_1 <- sd(coren_out_cor)
mu_2 <- mean(bet_out_cor)
sigma_2 <- sd(bet_out_cor)
normal_coren <- rnorm(10000,mu_1,sigma_1)
normal_bet <- rnorm(100000,mu_2,sigma_2)
Outer[,7] <- paste0("(",density(normal_coren)$x,",",density(
    normal_coren)$y,")")
Outer[,8] <- paste0("(",density(normal_bet)$x,",",density(normal_bet)
    $y,")")
write.csv(Outer,file="Kernels.csv",row.names=FALSE)
```

Script E.15: R implementation of the homophily rearrangement solution stability and solution correlation experiment

E.7.2 Multivariate homophily rearrangement

```
Rho ← c(0.5,0,-0.5)
for (r in Rho){
  Out ← c()
  for (i in 1:1000){
    Sigma ← matrix(1,2,2)
    Sigma[1,2] ← Sigma[2,1] ← r
    mu ← c(0,0)
    X ← data.frame(mvrnorm(n = 100, mu, Sigma, tol = 1e-6))
    W← Weightgenerator(10)[[1]]
    Phi ← c(-0.5,-0.5)
    t ← randomkick_generator_multi(Phi,X,W)
    Out[i] ← t
    print(t)
  }
  write.csv(Out,file=paste0("./Negative/Correlation_Test_Phi_",Phi
      [1],"_Rho_",r,".csv"),row.names=FALSE)
}
```

Script E.16: R implementation of the correlated features – solution time experiment

```
merging_up ← function(path_to_files,output_name){
    docs ← list.files(path_to_files)
    proper_frame ← data.frame(matrix(0,512,3))
    i ← 0
    for (doc in docs){
        i ← i + 1
        proper_path ← paste0(path_to_files,doc)
        seri ← read.csv(proper_path)
        proper_frame[,i] ← paste0("(",density(seri$x)$x,",",density(
            seri$x)$y,")")
        colnames(proper_frame)[i]← strsplit(strsplit(doc,"Test_")
            [[1]][2],".csv")[[1]][1]
    }
    write.csv(proper_frame,file=output_name,row.names = FALSE)
}

path_to_files ← "./Negative/"
output_name ← "Negative_Distributions.csv"
merging_up(path_to_files,output_name)

path_to_files ← "./Positive/"
output_name ← "Positive_Distributions.csv"
merging_up(path_to_files,output_name)
```

Script E.17: R implementation of the correlated features – solution time data aggregation

```
Rho ← c(-0.5,-0.4,-0.3,-0.2,-0.1,0,0.1,0.2,0.3,0.4,0.5)
Dimensions ← c(10,12,14,16)
Means_out ←data.frame(matrix(0,4,11))
Medians_out ←data.frame(matrix(0,4,11))
i ← 0
for (d in Dimensions){
  i ← i +1
  j ← 0
  for (r in Rho){
    j ← j +1
    Vector ← read.csv(paste0("./Size_Rho/DIM_",d,"_Rho_",r,".csv"),
        stringsAsFactors = FALSE)
    Means_out[i,j] ←mean(Vector$x)
    Medians_out[i,j] ←median(Vector$x)
  }
}

Means_out ←data.frame(t(Means_out))
Medians_out ←data.frame(t(Medians_out))

for (i in 1:4){
  Means_out[,i+4] ← paste0("(",Rho,",",Means_out[,i],")")
  Medians_out[,i+4] ←  paste0("(",Rho,",",Medians_out[,i],")")
}
```

Script E.18: R implementation of the correlated features – system size experiment

```
Rho ← c(-0.5,0,0.5)
Phi ← c(-0.5,0.5)
for (r in Rho){
  for (p in Phi){
    for (i in 1:1000){
      print(i)
      Sigma ← matrix(1,2,2)
      Sigma[1,2] ← Sigma[2,1] ← r
      mu ← c(0,0)
      X ← data.frame(mvrnorm(n = 100, mu, Sigma, tol = 1e-6))
      W ← Weightgenerator(10)[[1]]
      Phi ← c(p,p)
      set.seed(i)
      Calvalues ← randomkick_generator_multi(Phi,X,W)
      write.csv(Calvalues, file = paste0("C:/Users/212551747/Desktop
          /Divergence/",r,"_",p,"/Run_Rho_",r,"_Phi_",p,"_number_",i,
          ".csv"), row.names = FALSE)
    }
  }
}
```

Script E.19: R implementation of the correlated features and homophily experiment

```
B <-list.dirs()[2:7]
for (i in 1:6){
    maxer <- 0
    docs <- list.files(B[i])
    for (doc in docs){
      DF <- read.csv(paste0(B[i],"/",doc),stringsAsFactors = FALSE)
      print(nrow(DF))
      if (nrow(DF)>maxer){maxer <- nrow(DF)}
    }
    Out_DF <-data.frame(matrix(as.numeric(strsplit(B[i],"_")[[1]][2])
        ,maxer,2000))
    j<-0
    for (doc in docs){
      j <- j + 1
      DF <- read.csv(paste0(B[i],"/",doc),stringsAsFactors = FALSE)
      control <- nrow(DF)
      m1 <- 2*j
      m2 <- 2*j-1
      Out_DF[1:control,m1:m2] <- DF
    }
    write.csv(Out_DF, file=paste0(strsplit(B[i],"./")[[1]][2],".csv")
        ,row.names=FALSE)
    Means <- apply(Out_DF,1,mean)
    t <- 1:length(Means)
    plot(t,Means)
}
```

Script E.20: R implementation of the convergence to the target homophily experiment

```
to_read_in ← list.files()[c(2,4,6,8,10,12)]
i ← 1
OUTF ← data.frame(matrix(0,2000,7))
OUTF[,1] ←1:2000
colnames(OUTF)[1] ← "Time"
for (doc in to_read_in){
  i ← i + 1
  DF ← read.csv(doc,stringsAsFactors = FALSE)
  Evalue ← apply(DF,1,mean)[1:2000]
  OUTF[,i] ← paste0("(",OUTF[,1],",",Evalue[1:2000],")")
  colnames(OUTF)[i] ←paste0("rho",doc,"phi")
  plot(OUTF[,1],Evalue)
}
write.csv(OUTF, file = "./Results/Impulses.csv",row.names=FALSE)
```

Script E.21: R code to aggregate the convergence to the target homophily experiment I.

```
to_read_in ← list.files()[c(2,4,6,8,10,12)]
i ← 1
OUTF ← data.frame(matrix(0,2000,7))
OUTF[,1] ←1:2000
colnames(OUTF)[1] ← "Time"
for (doc in to_read_in){
  i ← i + 1
  DF ← read.csv(doc,stringsAsFactors = FALSE)
  even ← (1:1000)*2
  odd ← even -1
  Left ← DF[,c(even)]
  Right ← DF[,c(odd)]
  Funk ← abs(Left-Right)
  Evalue ← apply(Funk,1,mean)[1:2000]
  OUTF[,i] ← paste0("(",OUTF[,1],",",Evalue[1:2000],")")
  colnames(OUTF)[i] ←paste0("rho",doc,"phi")
  plot(OUTF[,1],Evalue)
}
write.csv(OUTF, file = "./Results/Abs_Impulses.csv",row.names=FALSE)
```

Script E.22: R code to aggregate the convergence to the target homophily experiment II.

E.7.3 Similarity based diffusion

```
Phi ← c(-0.5,0,0.5)
i ← 0
W← Weightgenerator(50)[[1]]
Dis ← data.frame(matrix(0,5000,3))
for (p in Phi){
  i ← i + 1
  X ← rnorm(2500,0,1)
  Xhat ← randomkick_generator(p,X,W)
  Edges←get.edgelist(g)
  Dis[,i] ← abs(Xhat[Edges[,1]]-Xhat[Edges[,2]])
}
Dis←Dis[,1:3]
write.csv(Dis,file="Dis.csv",row.names=FALSE)
Gams ← c(1,0.5)
Pvalues ← c(0.9,0.45)
for (Gam in Gams){
  for (P_0 in Pvalues){
    Dis ← read.csv("Dis.csv",stringsAsFactors = FALSE)
    Dis$Het ← P_0* exp(-Gam*Dis[,1])
    Dis$Neutral ← P_0* exp(-Gam*Dis[,2])
    Dis$Hom ← P_0* exp(-Gam*Dis[,3])
    Out ←data.frame(matrix(0,512,3))
    colnames(Out) ← c("Het","Neutral","Hom")
    Out$Het ← paste0("(",density(Dis$Het)$x,",",density(Dis$Het)$y,"
        )")
    Out$Neutral ← paste0("(",density(Dis$Neutral)$x,",",density(Dis$
        Neutral)$y,")")
    Out$Hom ← paste0("(",density(Dis$Hom)$x,",",density(Dis$Hom)$y,"
        )")
    write.csv(Out,file=paste0("Distribution_P_",P_0,"_Gam_",Gam,"
        .csv"),row.names = FALSE)
  }
}
```

Script E.23: R implementation of the changing pairwise transmission probabilities experiment

```
similarity_based_diffusion <- function(g,X,gamma,P_0,randn,non_ran,
    Identifier){
  B <- c(1:vcount(g))
  if (randn == TRUE) {Btilde <- c(sample(B, 1))}else {Btilde <- c(
     non_ran )}
  if(X[Btilde]>0){seeder = 1}else{seeder=0}
  B <- setdiff(B, Btilde)
  t <- 0
  Timeout<-Yout<-c()
  X_inf_out<-X_noninf_out<-rep(1,1)
  Timeout[t+1] <- t
  N <- vcount(g)
  Yout[t+1] <- length(Btilde)/N
  X_inf_out[1] <- t(c(mean(X[Btilde,1])))
  X_noninf_out[1] <- t(c(mean(X[B,1])))
  while (N != length(Btilde) ){
    print(paste0("The time period is: ",t))
    print(paste0("The number of infected nodes: ", length(Btilde)))
    K <- c()
    t <- t + 1
    for (i in Btilde){
      I <- c()
      N_G_I <- neighborhood(g,1,(V(g)[i]))[[1]]
      for(j in N_G_I){
        P <- runif(1,0,1)
        if (ncol(X)==1){Pij <- P_0*exp(-gamma[i]*sum(abs(X[i]-X[j])))
           }
        else{Pij <- P_0*exp(-gamma[i]*sum(abs(X[i,]-X[j,])))}
        if(P < Pij){I <- union(I,j)}
      }
      K <- union(K,I)
    }
    B <- setdiff(B,K)
    Btilde <- union(Btilde,K)
    Timeout[t+1] <- t
    Yout[t+1] <- length(Btilde)/N
    X_inf_out <-rbind(X_inf_out, c(mean(X[Btilde,1])))
```

```
35      X_noninf_out ← rbind(X_noninf_out,c(mean(X[B,1])))
36    }
37    Out ←rbind(Timeout,Yout,t(X_inf_out),t(X_noninf_out))
38    write.csv(Out,file=paste0("./Results/Result",Identifier,"
          seederstate",seeder,".csv"),row.names = FALSE)
39  }
```

Script E.24: R implementation of the similarity based diffusion model used for the experiments

```
1  W← Weightgenerator(10)[[1]]
2  g ← graph_from_adjacency_matrix(W,mode="undirected")
3  Phi ← c(-0.8,0.01,0.8)
4  for (ph in Phi){
5       for (i in 1:100){
6       set.seed(i)
7       print(i)
8       X ← c(cbind(rnorm(vcount(g),0,1)))
9       if (ph != 0.01){
10      X ←randomkick_generator(ph,X,W)}
11      X ← cbind(X)
12      P_0 ← 0.8
13      randn ← TRUE
14      non_ran ← 100
15      gamma ← rep(0.1,vcount(g))
16      Identifier ← paste0("Lattice_Spread_BaseGamma_Phi_",ph,"Run_",i)
17      similarity_based_diffusion(g,X,gamma,P_0,randn,non_ran,
            Identifier)
18    }
19  }
```

Script E.25: R implementation of the similarity based diffusion model

```
library(igraph)
W<- Weightgenerator(10)[[1]]
g <- graph_from_adjacency_matrix(W,mode="undirected")
Phi <- c(-0.8,0.01,0.8)
P_values <- c(0.4,0.45,0.5,0.55,0.6,0.65,0.7,0.75,0.8)
for (ph in Phi){
  for (p in P_values){
    for (i in 1:200){
    set.seed(i)
    print(i)
    X <- c(cbind(rnorm(vcount(g),0,1)))
    if (ph != 0.01){
    X <-randomkick_generator(ph,X,W)}
    X <- cbind(X)
    P_0 <- p
    randn <- TRUE
    non_ran <- 100
    gamma <- rep(1,vcount(g))
    Identifier <- paste0("Lattice_Spread_Pbase",p,"Phi_",ph,"Run_",i)
    similarity_based_diffusion(g,X,gamma,P_0,randn,non_ran,
        Identifier)
    }
  }
}
```

Script E.26: R implementation of the changing baseline transmission probability and diffusion time experiment

```
library(igraph)
W<- Weightgenerator(10)[[1]]
g <- graph_from_adjacency_matrix(W,mode="undirected")
Phi <- c(-0.8,0.01,0.8)
Gammas <- c(1:8)/10
for (ph in Phi){
  for (G in Gammas){
    for (i in 1:200){
    set.seed(i)
    print(i)
    X <- c(cbind(rnorm(vcount(g),0,1)))
    if (ph != 0.01){
    X <-randomkick_generator(ph,X,W)}
    X <- cbind(X)
    P_0 <- 0.5
    randn <- TRUE
    non_ran <- 100
    gamma <- rep(G,vcount(g))
    Identifier <- paste0("Lattice_Spread_Gamma",G,"Phi_",ph,"Run_",i)
    similarity_based_diffusion(g,X,gamma,P_0,randn,non_ran,
        Identifier)
    }
  }
}
```

Script E.27: R implementation of the changing sensitivity to dissimilarity and diffusion time experiment

```
similarity_based_diffusion <- function(g,X,gamma,P_0,randn,non_ran,
    Identifier){
  B <- c(1:vcount(g))
  if (randn == TRUE) {Btilde <- c(sample(B, 1))}else {Btilde <- c(
     non_ran )}
  if(X[Btilde]>0){seeder = 1}else{seeder=0}
  B <- setdiff(B, Btilde)
  t <- 0
  Timeout<-Yout<-c()
  X_inf_out<-X_noninf_out<-rep(1,1)
  Timeout[t+1] <- t
  N <- vcount(g)
  Yout[t+1] <- length(Btilde)/N
  X_inf_out[1] <- t(c(mean(X[Btilde,1])))
  X_noninf_out[1] <- t(c(mean(X[B,1])))

  while (N != length(Btilde) ){

    print(paste0("The time period is: ",t))
    print(paste0("The number of infected nodes: ", length(Btilde)))
    K <- c()
    t <- t + 1
    for (i in Btilde){

      I <- c()
      N_G_I <- neighborhood(g,1,(V(g)[i]))[[1]]

      for(j in N_G_I){
        P <- runif(1,0,1)
        if (ncol(X)==1){Pij <- P_0*exp(-gamma[i]*sum(abs(X[i]-X[j])))
            }
        else{Pij <- P_0*exp(-gamma[i]*sum(abs(X[i,]-X[j,])))}
        if(P < Pij){I <- union(I,j)}
      }
      K <- union(K,I)
    }
    B <- setdiff(B,K)
```

```
35      Btilde <- union(Btilde,K)
36      Timeout[t+1] <- t
37      Yout[t+1] <- length(Btilde)/N
38      X_inf_out <-rbind(X_inf_out, c(mean(X[Btilde,1])))
39      X_noninf_out <- rbind(X_noninf_out,c(mean(X[B,1])))
40    }
41    Out <-rbind(Timeout,Yout,t(X_inf_out),t(X_noninf_out))
42    write.csv(Out,file=paste0("./Binary/Homophily/Results",Identifier,
          "seederstate",seeder,".csv"),row.names = FALSE)
43  }
```

Script E.28: R implementation of the similarity based diffusion model with discriminating seeders

```
1  Phi <-c(0.0001,0.8)
2  m <- 60
3  for (ph in Phi){
4      for (i in 1:500){
5        set.seed(i)
6        print(i)
7        X <- sample(c(0,1), vcount(g),prob=c(0.7,0.3), replace = TRUE)
8        X <- cbind(X)
9        P_0 <- 0.5
10       randn <- TRUE
11       non_ran <- 100
12       gamma <- rep(0.1,vcount(g))
13       gamma[X>0] <- m* gamma[X>0]
14       Identifier <- paste0("Lattice_Spread_BaseGamma",m,"Phi_",ph,"
             Run_",i)
15       similarity_based_diffusion(g,X,gamma,P_0,randn,non_ran,
             Identifier)
16     }
17 }
```

Script E.29: R implementation of the discriminative seeding of diffusion experiment