



UNIVERSITAT POLITÈCNICA DE CATALUNYA

BARCELONATECH

Facultat d'Informàtica de Barcelona



TITOL

SUBTITOL

Autor

Data defensa

Director

Department director

MASTER IN INNOVATION AND RESEARCH IN INFORMATICS
Especialitat

Abstract

Quantitative linguistics study human language using statistical methods. Its aim is to build general theory from the statistical laws observed in a wide variety of languages. As part of the scientific method, this theory should be able to make novel predictions. This thesis is based on a family of models of human language ([11] and generalized in [13]). These models have shown to reproduce language laws, such as Zipf's law. They have also been used to make predictions, such as predicting the biases present in child word learning. This family of models is based on the optimization of information theoretic measures in a bipartite graph. Two specific models are studied here, the internal model and the external model. The mathematics behind the models are derived and implemented in code. Further mathematical work is also presented, which is used in order to speed up the optimization process: instead of statically recalculating every measure in each step, the effect of the changes done in the step is calculated dynamically. This introduces further problems, such as numerical error from floating point operations, which is also addressed. An open source tool `nom?` has been developed in C++. This tool implements the family of models, as well as an optimization algorithm consisting of a Markov chain Monte Carlo method at zero temperature. This tool is shown to be both powerful and generic enough to replicate findings from previously studied models in this family. This thesis presents both the new tool which can hopefully be used by researchers to further investigate this family of models, as well as novel findings obtained through the use of this tool, such as a consistent relationship between word age and frequency.

Contents

1	Introduction	3
1.1	Background	5
1.1.1	Linguistic laws	5
1.1.2	Similar research	7
1.2	General goals	8
1.3	Introduction to the model	9
1.3.1	Bipartite graph	9
1.3.2	Information theory	9
1.3.3	The ϕ parameter	12
1.3.4	The internal model	12
1.3.5	The external model	12
1.3.6	Optimization	13
1.3.7	Dynamic and static equations	13
1.4	Goals	14
1.5	Hypothesis	14
1.6	Challenges	15
1.6.1	Quantitative linguistics	15
1.6.2	Computational	15
1.6.3	Dynamic calculation	15
1.7	Outline of the thesis	16
2	Model	17
2.1	Mathematical aspect	17
2.1.1	Common concepts	18
2.1.2	The internal model	18
2.1.3	The external model	23
2.2	Computational aspect	28
2.2.1	The internal model	29
2.2.2	The external model	30
3	Methods	32
3.1	Model implementation	32
3.2	Optimization	35
3.3	Parallelization	36

3.4	Verification	36
3.5	Numerical precision problems	37
3.6	Other problems	37
4	Results	38
4.1	Verification of previous results	38
4.1.1	Results from the internal model (2005)	38
4.1.2	Results from the external model (2003)	43
4.2	Results for current models	43
4.2.1	Information Theoretic	43
4.2.2	Inside selected lambda	62
4.2.3	Graph visualization (smaller graphs)	81
5	Discussion	90
5.1	mates (linguistica quantitativa)	90
5.1.1	zipf law en regressio lineal del log-log?	90
5.1.2	relacio edat-frequencia	90
5.2	computacional	90
5.2.1	minim local	90
5.3	future work	91
5.3.1	altres metodes d'optimitzacio	91
5.3.2	altres valors de phi	91
5.3.3	vocabulary nens segon model	91
5.3.4	evitar error numeric encara mes	91
A	Code	92
A.1	repo	92
A.2	Program parameters	92
B	Formulae	94
B.1	Properties	94
B.1.1	Simplification of the equations of entropies	94
B.1.2	Proof that $\sum_{i=1}^n \mu_i^\phi \chi_i = \rho$	95
B.2	Derivation of the joint entropy in the external model	95
B.3	Dynamic Equations for the external model	97

Chapter 1

Introduction

Human languages obey many regularities which have been studied and formalized into laws. One of the most well known such laws is Zipf's law, which states that

$$p(i) \sim i^{-\alpha} \quad (1.1)$$

where i is a word's rank. Generally, $\alpha \approx 1$. [22] This is the case not only for English but for every tested human language. [14] Figure 1.1 shows this relationship for the first 10 million words of Wikipedia dumps in 30 different languages. This relationship also holds for artificial languages like Esperanto. There have been many efforts to understand why this and other empiric laws occur consistently in human language.

This thesis focuses on a family of models [11] introduced initially to shed light into the origins of Zipf's law. [9] [12] The focus of this thesis are two models from this family, which were generalized with the addition of a new parameter ϕ . [13] They are referred to as the "internal model" and the "external model" throughout. The key difference between these two models is whether they consider meanings to follow a probability distribution *internal* or *external* to the model.

It has been argued [12] [22] that human language is result of attempting to minimize the effort of both the speaker and the hearer. Something that Zipf referred to as the *principle of least effort*. These models follow this principle and are based on the minimization of a cost function, which is calculated in terms of information theoretic measures.

The motivation for this thesis is that these models could describe other language laws. Zipf's law for word frequencies is quite well known but there are other patterns in human language. One such pattern is the meaning-frequency law, also first reported by Zipf in his book. [22] Another is the relationship between the age of a word and its frequency, also studied by Zipf. [22] The consistent appearance of this law in

In order to study these patterns in the model, a C++ program is implemented as part of this thesis. This program is flexible and powerful enough to replicate

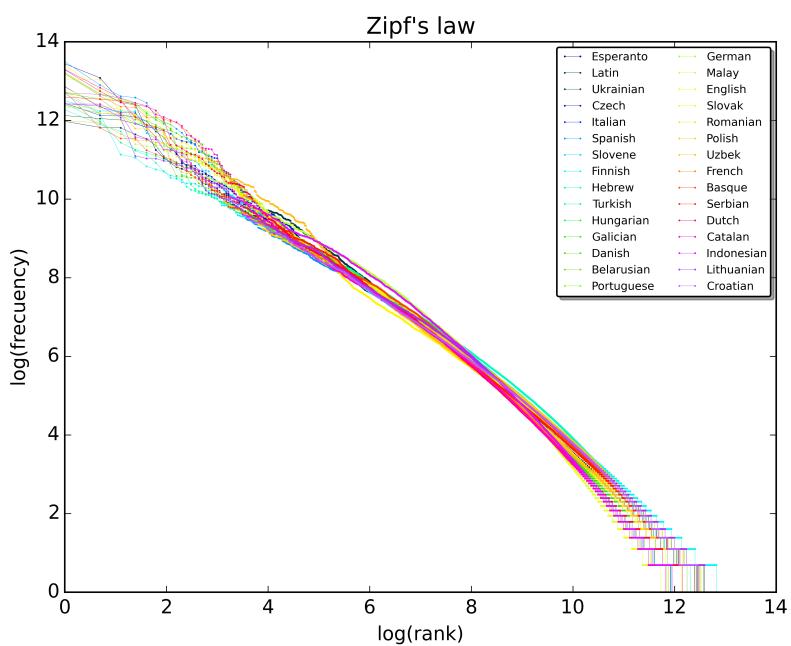


Figure 1.1: The relationship of the logarithm of a word's frequency versus its rank for 30 human languages including Esperanto, an artificial language. Data from 30 Wikipedia dumps from October 2015.

(Sergio Jimenez/Wikimedia, CC BY-SA 4.0 [19])

previous results [12] [9] and also capable of generating results for newer versions of these models. This program is one of the major contributions of this thesis. It is released under an open source license with the hope that it can help others study these models and similar ones.

The rest of this chapter is divided as follows. Section 1.1 revises the state of the art. Previous articles about this family of models along with their responses and critiques, until the more current models are reached. Section 1.2 outlines the set of goals of this thesis without entering into detail about the models. Section 1.3 introduces the family of models and the two models that are studied in detail. Having introduced the model, section 1.4 goes into more detail about the objectives of the thesis and section 1.5 covers the hypothesis. Section 1.6 explains the main challenges that from this work. Finally, Section 1.7 outlines the rest of the thesis.

1.1 Background

This section covers the “state of the art” or background of the thesis. Section 1.1.1 explains some of the linguistic laws that have been found in more detail. Section 1.1.2 explores similar approximations to general problems and other models that have been proposed to explain Zipf’s law, comparing them to these models.



1.1.1 Linguistic laws

Here some space is dedicated to describe the linguistic laws that are studied in the model in more detail.

Zipf’s word-frequency law

This law is well known as Zipf’s law. This thesis opens with its formulation in Equation (1.1) due to its importance. A word’s frequency is inversely proportional to its rank when ordered by frequency. The negative exponent α is typically around 1. [22]

This is one of the most well known laws in quantitative linguistics and previous models were already seen to follow it. [12] [9]

Zipf’s meaning-frequency and meaning distribution laws

Zipf’s meaning-frequency law is less well known. The law states the dependency between the number of meanings of a word μ and its frequency f , [22]

$$\mu \propto f^\delta.$$

Zipf derived the meaning-frequency law from his famous law of word-frequency, Equation (1.1) and from his law of meaning distribution,

$$\mu \propto i^{-\gamma}$$

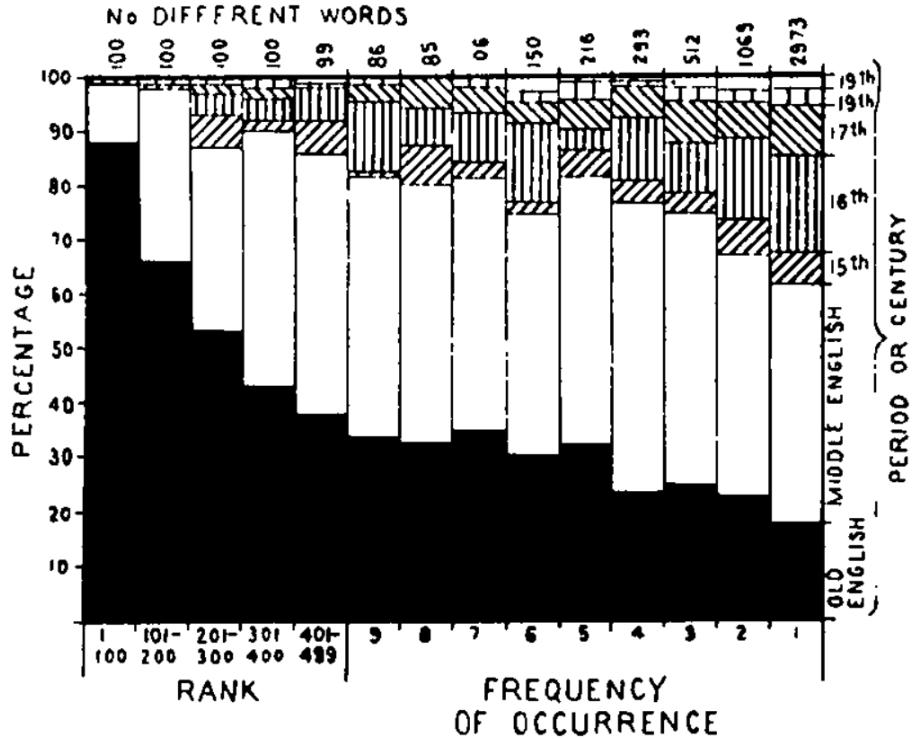


Figure 1.2: Age of a word as a function of its frequency. The right side of the y axis indicates the historical period or century when a word was introduced and the left side the percentage of words. Each of the colors and patterns of the columns in the graph correspond with a time period. As indicated in the x axis, the first five columns refer to the most common words by rank, while the next columns refer to the words with the specified frequency of occurrence.

where i is the rank of the word and $\gamma \approx 1/2$

As with the constant α , δ and γ can be estimated using a regression method from data.

Zipf argued, and it has been proven [7] that

$$\delta = \frac{\gamma}{\alpha}. \quad (1.2)$$

Zipf's Age-frequency law

Zipf argued in his book [22] that older words (words that have existed in a language for a longer amount of time) would be more frequent. This was tested empirically, as seen in Figure 1.2.

1.1.2 Similar research

This section goes over other problems that have been solved similarly and other models that have attempted to explain Zipf's law. A summary table between these alternative models and the models presented in this thesis is also shown.

Similar approximations to general problems

 There has been research which can be related to the work done in this thesis.

Terry Regier argued [17] that color naming in human language corresponded with optimal partitions of the color space. This is similar to the models presented here which, inspired by Zipf's *principle of least effort*, consider language to be the result of optimizing speaker and hearer effort. Regier's research showed that, indeed, color naming in many human languages was closely related to the optimal partitions of the color space. The idea of language being "optimal" is not new.

 In shortest path calculation algorithms, when a change takes place it is common to adjust the result based on the change instead of recalculating the entire process. These changes offer speed ups of several orders of magnitude. [2] This is similar to the static vs dynamic approach (Section 1.3.7). In this case, the dynamic algorithm offers a significant speedup that justifies issues such as the increase in mathematical complexity.

Other models that explain Zipf's Law

One of the most common explanations for Zipf's law is the random typing model.[4] As a short summary, this model states that if characters are typed in sequence with a random chance of typing in a word separator then Zipf's law is reproduced.

While it is true that the Miller's random typing shows behavior similar to Zipf's law, it is also a shallow explanation that misses many important details. To name a few issues, Miller's model assumes that words are independent from each other. However, in human language, words are very much dependent on each other. Random typing is often offered as a null hypothesis to the theory that Zipf's law comes from the *principle of least effort*. However, it has been shown [10] that random typing is an optimal coding process, which would make it not suitable as a null hypothesis either.

The random typing model would also not be able to explain other linguistic laws, such as the relationship between word age and frequency or the meaning distribution law, as it does not take into account the meanings nor the ages of words. It could also not make more complex predictions, such as the vocabulary learning biases observed in children.

Another model which is used to explain the occurrences of power laws in nature is Simon's model.[20] This model argues that these power laws appear as a result of the way the system is formed. In the case of language, constant addition of new words and addition of new instances of already existing words at a rate proportional to the number of instances of a word.

	Random Typing	Simon's Model	Internal model ($\phi = 0$)	External model ($\phi = 0$)	Internal model ($\phi \neq 0$)	External model ($\phi \neq 0$)
Rank-frequency law	Yes	Yes	Yes	Yes	Yes	Yes
Meaning distribution law	No	No	No	No	No	No
Age-frequency law	No	Unknown	Yes	Yes	Yes	Yes
Vocabulary learning bias	No	No	Yes [8]	Unknown	Yes [3]	Unknown

Table 1.1: A summary table comparing various models of human language. Columns show various models of human language: Random Typing and Simon's model. Then the two models studied in this thesis: internal and external model. The value of ϕ indicates whether it is the older version of the model (equivalent of the general model with $\phi = 0$) or the more general version with $\phi \neq 0$. Rows show various predictions that these models could or could not do. The vocabulary learning bias was shown to be predicted with the internal model and more recently with the more generic version of the model ($\phi \neq 0$)

This model is not as widely known as the random typing model. However, it is also not as powerful as the models presented in this thesis. Like the random typing model, Simon's model cannot explain the meaning distribution law as it does not take meaning into account. It also lacks the complexity to make predictions such as the vocabulary learning biases of children. However, it could perhaps explain or at least try to predict the age frequency law, as the ages of words could be tracked.

Table 1.1 shows a comparison between the two models seen here (random typing and Simon's model) and the two models that this thesis presents. The two models are further fleshed out in following sections.

1.2 General goals

Before introducing the models, this is a summary of the more general goals of this thesis. Section 1.4 offers a more specific set of goals within the context of the models now properly introduced.

The thesis aims to reproduce and verify the results of the previous models. As the newer models are generic, they should also be able to reproduce the older models. The new models should also be able to produce new results, which might be able to predict new linguistic laws that the older models could not.

Another goal is the creation of an open source program which can be used to test and replicate the results obtained.

1.3 Introduction to the model

This is a short introduction to the models presented in this thesis. In this section, mathematical definitions and notation common to both models is given. They will be used throughout the remainder of the thesis and specially throughout Chapter 3 where a full explanation of both models is given.

Section 1.3.1 presents the bipartite graphs that form the *skeleton* of these models. Section 1.3.2 presents the information theoretic aspects, common to both models. In Section 1.3.3 the role of the ϕ parameter is explained. Sections 1.3.4 and 1.3.5 cover the parts that unique to the internal and external model respectively. These are the *flesh*, each covering the *skeleton* in a different way. Section 1.3.6 gives an overview of the optimization process. Section 1.3.7 explains the reasoning and difference of the static and dynamic versions of the implemented algorithms.

1.3.1 Bipartite graph

Both models studied in this thesis are based on the idea of a bipartite graph. Bipartite graphs are comprised of two sets of elements and edges can only appear between an element of one set and an element of the other.

Our two sets are S and R . S is a set of size n containing all words. The notation s_i is used to refer to some element i of the set S . R is a set of size m containing all meanings. The notation r_j is used to refer to some element j of the set R .

The bipartite graph is represented using the adjacency matrix $A_{n,m}$ (or simply A) in most cases. $A_{n,m}$ is a $n \times m$ binary matrix representing whether an edge exists or not in the graph. Mathematically, each element of the matrix A is defined as

$$a_{i,j} = \begin{cases} 1 & \text{if there exists an edge between } s_i \in S \text{ and } r_j \in R \\ 0 & \text{otherwise.} \end{cases}$$

In some cases it is more convenient to represent the bipartite graph as the set E of all edges,

$$E = \{(s_i, r_j) \mid \text{there exists an edge between } s_i \in S \text{ and } r_j \in R\}.$$

For brevity, the pair (s_i, r_j) is often represented as (i, j) .

The degree of the word i is given as μ_i , while the degree of a meaning j is given as ω_j .

Figure 1.3 displays an example of a simple bipartite graph. A graphical representation is included, as well as the corresponding values of the mathematical concepts discussed in this section up to here.

1.3.2 Information theory

Information theory measures are used to compute the cost function of the optimization process. Here we go through a short overview of information theory

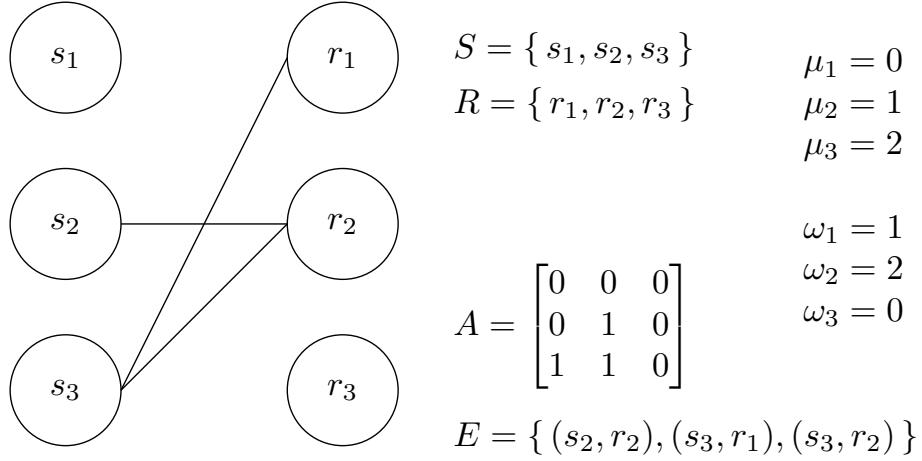


Figure 1.3: A bipartite graph with the corresponding adjacency matrix A , set of edges E and vertex degrees μ and ω for each vertex.

formulas. They can be found in [5].

The general formula for the entropy of a set X is

$$H(X) = \sum_{x \in X} p(x) \log p(x) \quad (1.3)$$

where $p(x)$ is the probability associated with the element x . The general formula for the joint entropy of two sets X and Y is defined as

$$H(X, Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log p(x, y) \quad (1.4)$$

where $p(x, y)$ is the joint probability of the elements x and y .

Any other information theoretic measures concerning two sets can be obtained from $H(X)$, $H(Y)$ and $H(X, Y)$. Mutual information

$$I(X, Y) = H(X) + H(Y) - H(X, Y),$$

and conditional entropies

$$H(X|Y) = H(X, Y) - H(Y)$$

and

$$H(Y|X) = H(X, Y) - H(X).$$

The diagram on Figure 1.4 shows these relationships in a more visual way.

Based on Equation (1.3), we define the entropies of words and meanings, $H(S)$ and $H(R)$ respectively, as

$$H(S) = \sum_{i=1}^n p(s_i) \log p(s_i) \quad (1.5)$$

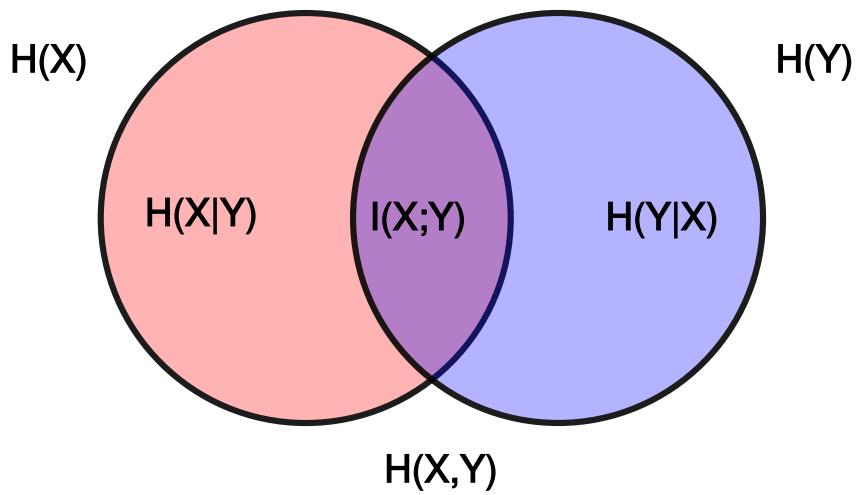


Figure 1.4: This diagram illustrates the relationships between information theoretic measures for two sets X and Y . The area occupied by both left and right circle is the joint entropy $H(X, Y)$. The circle on the left (both the red and the violet areas) represents the marginal entropy $H(S)$ while the circle on the right (both blue and violet areas) represents the marginal entropy $H(Y)$. The red area is the conditional entropy $H(X|Y)$ while the blue area represents the conditional entropy $H(Y|X)$. The violet area represents the mutual information $I(X, Y)$.

and

$$H(R) = \sum_{j=1}^m p(r_j) \log p(r_j). \quad (1.6)$$

We also define the joint entropy of words and meanings, from Equation (1.4), as

$$H(S, R) = \sum_{i=1}^n \sum_{j=1}^m p(s_i, r_j) \log p(s_i, r_j). \quad (1.7)$$

1.3.3 The ϕ parameter

The ϕ parameter, which has already appeared up to this point, is used to generalize the older versions of the two main models studied.

In [9] and [12] these two models were introduced. Later, the ϕ parameter is added to generalize the model and hopefully be able to predict more linguistic laws. [13]

As is presented in the following sections, when $\phi = 0$, the older models are retrieved from the newer ones.

1.3.4 The internal model

In this model, the joint probability of a word s_i and a meaning r_j is proportional to the product of their degrees to the ϕ power. When $\phi = 0$, the joint probability depends only on whether there is a connection between the word and the meaning. Mathematically,

$$p(s_i, r_j) \propto a_{i,j}(\mu_i \omega_j)^\phi.$$

When $\phi = 0$ we recover a previous simpler model. [9]

The actual joint probability is derived from Equation (1.3.4), and from it the marginal probabilities of words and meanings are also derived. Ultimately this leads to the information theoretic equations seen in Section 1.3.2.

1.3.5 The external model

This model was introduced in [12] with meaning probabilities being constant and disconnected meanings being disallowed. Here we present a generalization of the previous model. In this model, the probability of a meaning r_j is given *a priori* as $\pi(r_j)$. It is possible for a meaning to be disconnected from any words, in which case $p(r_j) = 0$. In [12] disconnected meanings were disallowed and $\pi(r_j) = \frac{1}{m}$. Additionally, $\phi = 0$ in Equation (1.9). Here $\pi(r_j)$ can follow any probability distribution and disconnected meanings are allowed. Zero values of $\pi(r_j)$ are disallowed, however, as meanings that could never occur are not the target of communication.

We then define $p(r_j)$ as

$$p(r_j) \propto (1 - \delta_{\omega_j, 0})\pi(r_j) \quad (1.8)$$

where $\delta_{a,b}$ is the Kronecker delta,

$$\delta_{a,b} = \begin{cases} 0 & \text{if } a \neq b \\ 1 & \text{if } a = b. \end{cases}$$

The word probability is then defined as the conditional probability of choosing a word given that a meaning has been chosen,

$$p(s_i|r_j) \propto a_{i,j} \mu_i^\phi. \quad (1.9)$$

With the marginal meaning probability and the conditional probability of a word given a meaning, the joint probability and marginal word probability can be calculated. From this the information theoretic equations are obtained.

1.3.6 Optimization

As seen in previous sections and will be seen in Section 1.5, our hypothesis is that the empirical laws observed in human language are the result of minimizing the effort of speaker and hearer.

This effort is defined in terms of information theoretic measures. We choose two competing forces. $H(S)$, the entropy of the words, and $I(S, R)$, the mutual information between words and meanings. It is the goal of any communications system to maximize $I(S, R)$. The smaller $I(S, R)$, the greater the effort for the hearer. The higher the entropy of words, however, they harder they are to access. When $H(S) = 0$, only a single word has probability 1 while all others are 0, meaning that knowing which word to choose is trivial, while $H(S)$ is maximum when all words have equal probability, giving no indication of which should be chosen.

We define $\Omega(\lambda)$ as the cost function we aim to minimize,

$$\Omega(\lambda) = -\lambda I(S, R) + (1 - \lambda)H(S) \quad (1.10)$$

where $0 \leq \lambda \leq 1$ is used to indicate the weight given to each of the two forces. When $\lambda = 0$, the minimization of $H(S)$ is completely favored while when $\lambda = 1$ the maximization of $I(S, R)$ is. When $\lambda = 1/2$, they are equally favored.

The optimization process follows a Markov chain Monte Carlo method at zero temperature. It consists of making mutations to A , changing $a_{i,j}$ from 1 to 0 or from 0 to 1. If a set of mutations results in a decrease of Ω , they are kept. Otherwise they are undone and another set is attempted.

1.3.7 Dynamic and static equations

As is seen later on, in Chapter 2, two sets of equations are derived for the information theoretic equations necessary to calculate Ω (Equation (1.10)).

A set of static equations recalculate everything from scratch. The set of dynamic equations calculate the changes done only after a mutation to A takes place.

As seen in Section 1.1.2 this dynamic recalculation can be more efficient. In the case of this thesis, small changes are continuously done to the graph, making several mutations to A then reevaluating Ω . Dynamically accounting for the differences brought about for these changes is much more efficient than recalculating every single measure each time.

Dynamic calculation, however, brings its own set of problems, such as the increase in mathematical complexity and the introduction of greater floating point error. Section 1.6.3 goes over this in more detail while Section 3.1 covers the measures taken in order to deal with these problems.

Table 2.1 in Chapter 2 shows the computational costs of the two models for specific cases that are implemented in the C++ open source program.

1.4 Goals

With the model introduced, more specific goals are stated. See also the overall goals in section 1.2.

- To create an open source tool implementing the general models.
 - This tool should be both powerful and flexible, allowing to replicate previous results for the case $\phi = 0$.
 - It should also be efficient and fast enough that results with relatively large graphs can be obtained in a reasonable time. This means using dynamic equations.
 - It should be relatively easy to use such that anyone who wishes to test these models by themselves can do it.
- Find out whether local minima exist. It is not clear whether there exists local minima for the case $\phi \neq 0$. The previous models where $\phi = 0$ have been analyzed mathematically. [18] [16] The case $\phi \neq 0$ is more complex and would be very hard to analyze mathematically to this level. It is possible that the optimization stops at a point where the function could still be minimized further but it is hard to find the set of mutations that minimize it. Or it could be that the function does reach a local minima.

1.5 Hypothesis

Several hypothesis are stated here.

As seen in section 1.1 and in [13], there is a relationship between the exponents of the Zipf's word-frequency and meaning-frequency laws. See Equation (1.2). The main hypothesis of the thesis is that linguistic laws appear due to an optimization process. It is under this assumption that the entire model works and the results are obtained. In the model we also neglect any effects of social interaction, which are the basis of other approaches to investigating human language such as the naming game. [1] Another hypothesis is that the optimization process does not necessarily arrive at a local minimum.

1.6 Challenges



The main challenges presented by this thesis are outlined here. They are divided into three categories: challenges relating to quantitative linguistics in Section 1.6.1, computational challenges in Section 1.6.2 and the challenges brought by the use of a dynamic calculation technique, in section 1.6.3.



1.6.1 Quantitative linguistics

The challenges relating to quantitative linguistics come from the fact that current models already can make some predictions. Can these new models predict anything new? Can they still make the predictions that the old models already could? Specifically, the $\phi = 0$ models presented in [9] and [12] already showed the appearance of Zipf's law of word frequencies. Can the new models with $\phi \neq 1$ (and more specifically $\phi = 1$) make any new predictions such as the law of meaning frequency, or the one relating word ages and frequencies? Can they still predict Zipf's law?

1.6.2 Computational

The computational challenges relate to the implementation of the model.

Verification is a very important step. The mathematics are complex, specially in the dynamic case, and a single mistake can invalidate all the results. Therefore, the implementation must be verified thoroughly and completely.

The stop condition of the optimization process is also difficult to properly implement. The algorithm should stop once it can be reasonably sure that a minimum has been reached. However, due to the size search space, it is unfeasible to check every possible set of mutations.

An important challenge is recalculating Ω efficiently enough that results with relatively big n and m can be executed in a reasonable time. The calculation of Ω directly influences the runtime of the program. In order to speed it up, dynamic calculations are used. Dynamic calculation, however, presents its own set of challenges.

1.6.3 Dynamic calculation

Dynamic calculation is much more mathematically complex than static calculation. This complexity makes the likelihood of a bug being introduced much more likely. Therefore the verification process must be even more thorough.

In addition to complexity, dynamic calculation implies many addition and subtraction operations. These operations introduce greater floating point error than others (such as multiplication and division). If the floating point error accumulates, it can change the result into a different one from the static method. Dynamic and static approaches should lead to identical results.

1.7 Outline of the thesis

The rest of the thesis is organized as follows:



Chapter 2 gives more detail of the models, and all the mathematical detail concerning the models (although long derivations or simple proofs that are otherwise unrelated are left for Appendix B). It is during this chapter that all the mathematics that are eventually implemented into the open source tool are showcased. The computational aspects are also seen here, showing the computational complexity of implementing these formulas. This chapter starts from the initial concepts seen in Section 1.3.



Chapter 3 goes into the implementation of the model. It includes full descriptions of various implementation details, of the optimization algorithm, the decisions regarding parallelization of the code and the approach to verification. It also covers some challenges unique to the implementation of the model and how they were resolved.

Chapter 4 introduces the results obtained from the open source tool. It includes the figures from the previous models [9] and [12] implemented with $\phi = 0$. It also includes new data obtained by setting $\phi = 1$.

Finally, Chapter 5 discusses the obtained results and their relation to certain linguistic laws that have been seen in this introduction. Future work is also discussed in this chapter, including alternative optimization methods.

Chapter 2

Model

This chapter covers the details about the two models previously introduced in Chapter 1, where a high level introduction to both the internal model and the external model was given. It follows a top down approach. General concepts from both types of model are outlined first and the specific details from each model are given afterwards.

As a reminder of the previous chapter a very brief definition of the two models follows. Both models are quite similar, they represent a way to connect words with meanings. Both meanings and words are given probabilities of being used, from which information theoretic measures are derived, which are ultimately used to obtain a cost function. The high level differentiating trait is the way in which the probability of a meaning is obtained. In the first studied model (internal model), the probabilities of both words and meanings are *internal* to the model, they only depend on the structure of the bipartite graph. In the second model (external model), the probabilities of meanings are *external* to the model, they come from a distribution of *a priori* probabilities.

The remainder of this chapter is divided into two sections. Section 2.1 deals with the mathematical definitions of the models and it is more theoretical in nature. Section 2.2 outlines the computational aspects of the two models and it is more practical and implementation oriented. This chapter does not deal with any actual details of the implementation of the model. These details are given in Chapter 3.

Section 1.3 from Chapter 1 introduces the the graph theory and information theory concepts that will appear during the rest of this chapter.

2.1 Mathematical aspect

This section covers the mathematical definition of the model. Section 2.1.1 introduces concepts common to both models. These concepts are then extended for either the internal model or the external model in Sections 2.1.2 and 2.1.3 respectively.

2.1.1 Common concepts

Here a few more common concepts that were not discussed in Chapter 1 are introduced.

As a reminder, the notation defined for the degree of the vertices in the graph is repeated. The word s_i has degree μ_i , while the meaning r_j has degree ω_j . μ and ω are defined as

$$\mu_i = \sum_{j=1}^m a_{i,j} \quad (2.1)$$

and

$$\omega_j = \sum_{i=1}^n a_{i,j}. \quad (2.2)$$

There is an additional parameter of the models, ϕ , which is not directly related to the bipartite graph. This parameter appears in both models and it is used to “emphasize” the effect of the vertex degrees in the calculations. As explained in previous sections, this parameter is a new addition to the models originally presented in [9] and [12].

With the parameter ϕ , the definitions of μ and ω can be generalized. μ_ϕ and ω_ϕ are defined as

$$\mu_{\phi,i} = \sum_{j=1}^m a_{i,j} \omega_j^\phi \quad (2.3)$$

and

$$\omega_{\phi,i} = \sum_{i=1}^n a_{i,j} \mu_i^\phi. \quad (2.4)$$

It can be seen that when $\phi = 0$ Equation (2.3) becomes Equation (2.1) and Equation (2.4) becomes Equation (2.2). In other words, $\mu_{0,i} = \mu_i$ and $\omega_{0,j} = \omega_j$

2.1.2 The internal model

Here the concepts outlined in Section 1.3.4 are worked out into actual probabilities and information theoretic equations.

This model is defined by the joint probability of a word s_i and a meaning r_j seen in Equation (1.3.4). As seen in Section 1.3.4, this probability must be proportional to the product of the degrees of s_i and r_j .

From this definition equations for the marginal probabilities are derived, which in turn are used to obtain the information theory expressions in the cost function Ω (Equation (1.10)). From these expressions, dynamic equations are derived to obtain the change experienced by the entropies after a single mutation has occurred on the adjacency matrix A .

Joint Probability

Adding a normalizing factor M_ϕ , the joint probability becomes

$$p(s_i, r_j) = \frac{1}{M_\phi} a_{i,j} (\mu_i \omega_j)^\phi. \quad (2.5)$$

This normalizing factor is obtained by applying the definition of probability

$$\sum_{i=1}^n \sum_{j=1}^m p(s_i, r_j) = 1.$$

Then,

$$M_\phi = \sum_{i=1}^n \sum_{j=1}^m a_{i,j} (\mu_i \omega_j)^\phi$$

or equivalently

$$M_\phi = \sum_{(i,j) \in E} (\mu_i \omega_j)^\phi. \quad (2.6)$$

Notably, when $\phi = 0$, M_ϕ is simply M and as follows from Equation (2.6), it's the total number of edges in the graph.

Marginal Probabilities

Formulas for the marginal probabilities $p(s_i)$ and $p(r_j)$ are derived from the joint probability using the general formula

$$p(x) = \sum_{y \in Y} p(x, y). \quad (2.7)$$

Applying Equation (2.7),

$$p(s_i) = \frac{\mu_i^\phi}{M_\phi} \sum_{j=1}^m a_{i,j} \omega_j^\phi.$$

Recall Equation (2.3),

$$p(s_i) = \frac{\mu_i^\phi \mu_{\phi,i}}{M_\phi}. \quad (2.8)$$

Symmetrically applying Equation (2.7) to obtain $p(r_j)$ and applying Equation (2.4) instead we obtain

$$p(r_j) = \frac{\omega_j^\phi \omega_{\phi,j}}{M_\phi}. \quad (2.9)$$

Entropies

$H(S, R)$ is obtained by applying the joint probability given in Equation (2.5) to Equation (1.7),

$$H(S, R) = - \sum_{i=1}^n \sum_{j=1}^m \frac{1}{M_\phi} a_{i,j} (\mu_i \omega_j)^\phi \log (a_{i,j} (\mu_i \omega_j)^\phi). \quad (2.10)$$

Property 1

This expression can be refined taking advantage of the equality

$$-\sum_i \frac{x_i}{T} \log \frac{x_i}{T} = \log T - \frac{1}{T} \sum_i x_i \log x_i.$$

This equality holds as long as

$$T = \sum_i x_i.$$

While is quite simple to prove that this is true, the derivation is given in Appendix B.1.1.

Applying Property 1 to Equation (2.10) we obtain

$$H(S, R) = \log M_\phi - \frac{\phi}{M_\phi} \sum_{i=1}^n \sum_{j=1}^m a_{i,j} ((\mu_i \omega_j)^\phi \log \mu_i \omega_j)$$

using E in the summation and adopting the convention that $0 \log 0 = 0$

$$H(S, R) = \log M_\phi - \frac{\phi}{M_\phi} \sum_{(i,j) \in E} (\mu_i \omega_j)^\phi \log \mu_i \omega_j \quad (2.11)$$

is reached.

The marginal word entropy is found applying the marginal probability in Equation (2.8) to the definition of $H(S)$ in Equation (1.5)

$$H(S) = - \sum_{i=1}^n \frac{\mu_i \mu_{\phi,i}}{M_\phi} \log \frac{\mu_i \mu_{\phi,i}}{M_\phi}.$$

Property 1 can be applied immediately, obtaining

$$H(S) = \log M_\phi - \frac{1}{M_\phi} \sum_{i=1}^n \mu_i^\phi \mu_{\phi,i} \log \mu_i^\phi \mu_{\phi,i}. \quad (2.12)$$

The convention $0 \log 0 = 0$ is applied here for disconnected words (where $\mu_i = 0$).

The marginal meaning entropy is found symmetrically, the marginal probability (Equation (2.9)) is applied to the definition of $H(R)$ (Equation (1.6)). With Property (1), we obtain

$$H(R) = \log M_\phi - \frac{1}{M_\phi} \sum_{j=1}^m \omega_j^\phi \omega_{\phi,j} \log \omega_j^\phi \omega_{\phi,j}. \quad (2.13)$$

also using the convention $0 \log 0 = 0$ for disconnected meanings ($\omega_j = 0$).

Dynamic Equations

The full derivation of the dynamic equations is given in [3]. Here only the final expressions of the dynamic equations are given.

Starting with compact expressions of the entropies (Equations (2.11), (2.12) and (2.13))

$$H(S, R) = \log M_\phi - \frac{\phi}{M_\phi} X(S, R) \quad (2.14)$$

$$H(S) = \log M_\phi - \frac{1}{M_\phi} X(S) \quad (2.15)$$

$$H(R) = \log M_\phi - \frac{1}{M_\phi} X(R) \quad (2.16)$$

with

$$X(S, R) = \sum_{(i,j) \in E} x(s_i, r_j) \quad (2.17)$$

$$X(S) = \sum_{i=1}^n x(s_i) \quad (2.18)$$

$$X(R) = \sum_{j=1}^m x(r_j) \quad (2.19)$$

$$x(s_i, r_j) = (\mu_i \omega_j)^\phi \log \mu_i \omega_j \quad (2.20)$$

$$x(s_i) = \mu_i^\phi \mu_{\phi,i} \log \mu_i^\phi \mu_{\phi,i} \quad (2.21)$$

$$x(r_j) = \omega_j^\phi \omega_{\phi,j} \log \omega_j^\phi \omega_{\phi,j}. \quad (2.22)$$

A prime mark is used to indicate a new value of a certain variable after a mutation has taken place. A variable without a prime mark indicates the value before the mutation took place. Suppose that $a_{i,j}$ mutates. Then

$$a'_{i,j} = 1 - a_{i,j} \quad (2.23)$$

$$\mu'_i = \mu_i + (-1)^{a_{i,j}} \quad (2.24)$$

$$\omega'_j = \omega_j + (-1)^{a_{i,j}} \quad (2.25)$$

We define the set of neighbors of any word s_i

$$\Gamma_S(i) = \{ r_j \mid (s_i, r_j) \in E \}, \quad (2.26)$$

and similarly the set of neighbors of any meaning r_j

$$\Gamma_R(j) = \{ s_i \mid (s_i, r_j) \in E \}. \quad (2.27)$$

Then, for any k such that $1 \leq k \leq n$, we have that

$$\mu'_{\phi,k} = \begin{cases} \mu_{\phi,k} - a_{ij}\omega_j^\phi + (1-a_{ij})\omega'_j{}^\phi & \text{if } k = i \\ \mu_{\phi,k} - \omega_j^\phi + \omega'_j{}^\phi & \text{if } k \in \Gamma_R(j) \text{ and } k \neq i \\ \mu_{\phi,k} & \text{otherwise.} \end{cases} \quad (2.28)$$

Likewise, for any l such that $1 \leq l \leq m$, we have that

$$\omega'_{\phi,l} = \begin{cases} \omega_{\phi,l} - a_{ij}\mu_i^\phi + (1-a_{ij})\mu'_i{}^\phi & \text{if } l = j \\ \omega_{\phi,l} - \mu_i^\phi + \mu'_i{}^\phi & \text{if } l \in \Gamma_S(i) \text{ and } l \neq j \\ \omega_{\phi,l} & \text{otherwise.} \end{cases} \quad (2.29)$$

These variables can be applied to Equations (2.20), (2.21) and (2.22) directly to obtain their values.

We define the set $E_{i,j}$ as the set of all edges connecting s_i and r_j with their neighbors. That is,

$$E_{i,j} = \{ (i, l) \mid l \in \Gamma_S(i) \} \cup \{ (k, j) \mid k \in \Gamma_R(j) \} \quad (2.30)$$

With these definitions, we can now obtain the expressions of $X'(S, R)$ from $X(S, R)$ and of M'_ϕ from M_ϕ .

$$\begin{aligned} M'_\phi = M_\phi - & \left[\sum_{(k,l) \in E_{i,j}} (\mu_k \omega_l)^\phi \right] - a_{ij}(\mu_i \omega_j)^\phi \\ & + \left[\sum_{(k,l) \in E_{i,j}} (\mu'_k \omega'_l)^\phi \right] + (1-a_{ij})(\mu'_i \omega'_j)^\phi. \end{aligned}$$

Similarly, the new value of $X(S, R)$ will be

$$\begin{aligned} X'(S, R) = X(S, R) - & \left[\sum_{(k,l) \in E_{i,j}} x(s_k, r_l) \right] - a_{ij}x(s_i, r_j) \\ & + \left[\sum_{(k,l) \in E_{i,j}} x'(s_k, r_l) \right] + (1-a_{ij})x'(s_i, r_j). \end{aligned} \quad (2.31)$$

The expressions of $X'(S)$ from $X(S)$ and $X'(R)$ from $X(R)$ used to dynamically update Equations (2.31), (2.32) and (2.33) are

$$X'(S) = X(S) - \left[\sum_{k \in \Gamma_R(j)} x(s_k) \right] - a_{ij}x(s_i) + \left[\sum_{k \in \Gamma_R(j)} x'(s_k) \right] + (1 - a_{ij})x'(s_i) \quad (2.32)$$

and

$$X'(R) = X(R) - \left[\sum_{l \in \Gamma_S(i)} x(r_l) \right] - a_{ij}x(r_j) + \left[\sum_{l \in \Gamma_S(i)} x'(r_l) \right] + (1 - a_{ij})x'(r_j). \quad (2.33)$$

Again, the full derivation of these equations is given in [3] and it is not included here to avoid repeating the same ideas.

The values of $H(S)$, $H(R)$ and $H(S, R)$ can be obtained by applying Equations (2.31), (2.32) and (2.33) to the definitions in Equations (2.14), (2.15) and (2.16) respectively.

Extreme cases and invariants

For verification purposes (see Section 3.4), the values of the entropies along with their invariants are also given here.

Extreme cases These cases can be easily derived from Equations (2.11), (2.12) and (2.13) ($H(S, R)$, $H(S)$ and $H(R)$ respectively) by assuming the corresponding condition and applying elementary algebra.

- For a single edge, $H(S), H(R), H(S, R) = 0$.
- For a complete graph, $H(S) = \log n, H(R) = \log m, H(S, R) = \log nm$
- For a one-to-one mapping of signals into meanings with $n = m, H(S), H(R), H(S, R) = \log n$

Invariants These invariants follow from basic information theory. [5]

- $0 \leq H(S) \leq \log n$
- $0 \leq H(R) \leq \log m$
- $0 \leq H(S, R) \leq \log nm$

2.1.3 The external model

Here the equations and concepts form 1.3.5 are worked out into probabilities and information theoretic equations.

In this model, $p(r_j)$ depends in an *a priori* probability $\pi(r_j)$ while $p(s_i|r_j)$ is defined in terms of μ_i^ϕ .

From the definition of probability,

$$\sum_{j=1}^m p(r_j) = 1, \quad (2.34)$$

even when there are disconnected meanings whose probability should be zero.

From Equation (1.8) and 2.34,

$$p(r_j) = \frac{(1 - \delta_{\omega_j,0})\pi(r_j)}{\rho} \quad (2.35)$$

where

$$\begin{aligned} \rho &= \sum_{j=1}^m (1 - \delta_{\omega_j,0})\pi(r_j) \\ &= 1 - \sum_{j=1}^m \delta_{\omega_j,0}\pi(r_j). \end{aligned} \quad (2.36)$$

The conditional probability of choosing a word given a meaning is proportional to the number of meanings associated with that word or zero if that meaning is not associated to the word.

Applying

$$\sum_{i=1}^n p(s_i|r_j) = 1$$

to Equation (1.9) and recalling Equation (2.4) we obtain

$$p(s_i|r_j) = \frac{a_{i,j}\mu_i^\phi}{\omega_{\phi,j}}. \quad (2.37)$$

The joint probability $p(s_i, r_j)$ is obtained by applying Equation (2.35) and Equation (2.37) to the definition

$$p(s_i, r_j) = p(s_i|r_j)p(r_j),$$

obtaining

$$p(s_i, r_j) = \frac{a_{i,j}(1 - \delta_{\omega_j,0})\mu_i^\phi\pi(r_j)}{\rho\omega_{\phi,j}}. \quad (2.38)$$

Marginal word probability

The probability of a word can be derived from Equation (2.38) and applying

$$p(s_i) = \sum_{j=1}^m p(s_i, r_j),$$

obtaining

$$p(s_i) = \frac{\mu_i^\phi \chi_i}{\rho} \quad (2.39)$$

with

$$\chi_i = \sum_{j=1}^m \frac{a_{i,j}(1 - \delta_{\omega_j,0})\pi(r_j)}{\omega_{\phi,j}}. \quad (2.40)$$

Entropies

Applying the definition of $H(R)$ (Equation (1.6)) to $p(s_j)$ (Equation (2.35)) we obtain

$$H(R) = - \sum_{j=1}^m \frac{(1 - \delta_{\omega_j,0})\pi(r_j)}{\rho} \log \frac{(1 - \delta_{\omega_j,0})\pi(r_j)}{\rho}.$$

Property 1 can be applied immediately to simplify $H(R)$

$$H(R) = \log \rho - \frac{1}{\rho} \sum_{j=1}^m (1 - \delta_{\omega_j,0})\pi(r_j) \log(1 - \delta_{\omega_j,0})\pi(r_j).$$

This is simplified further applying the convention $0 \log 0 = 0$

$$\begin{aligned} H(R) &= \log \rho - \frac{1}{\rho} \sum_{j=1}^m (1 - \delta_{\omega_j,0})\pi(r_j) \log \pi(r_j) \\ &= \log \rho + \frac{1}{\rho} \left(H_\pi(R) + \sum_{j=1}^m \delta_{\omega_j,0}\pi(r_j) \log \pi(r_j) \right) \end{aligned} \quad (2.41)$$

where $H_\pi(R)$ is the entropy of the *a priori* probabilities

$$H_\pi(R) = - \sum_{j=1}^m \pi(r_j) \log \pi(r_j)$$

$H(S, R)$ can be derived by applying Equation (2.38) to the information theory definition of $H(S, R)$ (Equation (1.7)). Alternatively, it can also be derived by using the equality

$$H(S, R) = H(S|R) + H(R).$$

This reduces the problem to finding $H(S|R)$ using

$$H(S|R) = \sum_{j=1}^m H(S|r_j)p(r_j)$$

and

$$H(S|r_j) = - \sum_{i=1}^n p(s_i|r_j) \log p(s_i|r_j).$$

Both approaches turn out to be quite cumbersome mathematically. For the sake of brevity they are not shown here and can be consulted in Appendix B.2. In either case, the resulting expression for $H(S, R)$ is

$$H(S, R) = \log \rho - \frac{1}{\rho} \sum_{j=1}^m (1 - \delta_{w_j, 0}) \pi(r_j) \left[\frac{\phi \nu_j}{\omega_{\phi, j}} + \log \frac{\pi(r_j)}{\omega_{\phi, j}} \right] \quad (2.42)$$

with

$$\nu_j = \sum_{i=1}^n a_{i,j} \mu_i^\phi \log(\mu_i). \quad (2.43)$$

$H(S)$ is derived quite easily by applying Equation (2.39) to the information theory definition (Equation (1.5))

$$H(S) = - \sum_{i=1}^n \frac{\mu_i^\phi \chi_i}{\rho} \log \frac{\mu_i^\phi \chi_i}{\rho}$$

Property 2

It is quite simple to see that

$$\sum_{i=1}^n \mu_i^\phi \chi_i = \rho,$$

however this might not be immediately obvious as it has not appeared at any point until now. Appendix B.1.2 shows that this property indeed holds.

Following Property 2, Property 1 can be applied here, obtaining

$$H(S) = \log \rho - \frac{1}{\rho} \sum_{i=1}^n \mu_i^\phi \chi_i \log \mu_i^\phi \chi_i \quad (2.44)$$

Dynamic Equations

The full derivation of the dynamic equations is given in Appendix B.3. Here only the final expressions of the dynamic equations are given. Recall Section 2.1.2 for many useful definitions. In all dynamic equations, a mutation on $a_{i,j}$ is assumed.

Compact expressions of the entropies (Equations (2.41), (2.42) and (2.44))

$$H(S, R) = \log \rho - \frac{1}{\rho} X(S, R) \quad (2.45)$$

$$H(S) = \log \rho - \frac{1}{\rho} X(S) \quad (2.46)$$

$$H(R) = \log \rho - \frac{1}{\rho} X(R) \quad (2.47)$$

with

$$X(S, R) = \sum_{j=1}^m (1 - \delta_{\omega_j, 0}) x(r_j) \quad (2.48)$$

$$X(S) = \sum_{i=1}^n x_{s_i} \quad (2.49)$$

$$X(R) = \sum_{j=1}^m (1 - \delta_{\omega_j, 0}) \pi(r_j) \log \pi(r_j) \quad (2.50)$$

$$x(s_i) = \mu_i^\phi \chi_i \log (\mu_i^\phi \chi_i) \quad (2.51)$$

$$x(r_j) = \pi(r_j) \left[\frac{\phi \nu_j}{\omega_{\phi, j}} + \log \frac{\pi(r_j)}{\omega_{\phi, j}} \right]. \quad (2.52)$$

Equations (2.23) ($a'_{i,j}$) and (2.24) (μ'_i) remain the same as in the internal model. Equation (2.25) (ω'_j) is unused in this model.

Equation (2.29) ($\omega'_{\phi, l}$ for any l such that $1 \leq l \leq m$) remains identical, while Equation (2.28) ($\mu'_{\phi, k}$ for any k such that $1 \leq k \leq n$) is unused in this model.

Recall as well the definition of the set of all edges to neighbors of s_i and r_j (Equation (2.30)). Two additional sets are defined for ease in the definition of these equations, $A_{i,j}(k)$ is the set of all meanings that are neighbors of i ($\Gamma_S(i)$) (plus the meaning r_j if not already included) that are also neighbors of k

$$A_{i,j}(k) = (\Gamma_S(i) \cup \{ r_j \}) \cap \Gamma_S(k). \quad (2.53)$$

The other set, $B_{i,j}(k)$ is the set of all words s_k such that $A_{i,j}(k)$ is not the empty set. The word s_i is always included,

$$B_{i,j}(k) = \{ s_k \mid A_{i,j}(k) \neq \emptyset \} \cup \{ s_i \} \quad (2.54)$$

The one mutation change equations for ρ , ν and χ are

$$\rho' = \rho - \delta_{\omega'_j, 0} \pi(r_j) + \delta_{\omega_j, 0} \pi(r_j), \quad (2.55)$$

$$\nu'_l = \begin{cases} \nu_l + (1 - a_{ij}) \mu'_i \log \mu'_i - a_{ij} \mu_i^\phi \log \mu_i & \text{if } l = j \\ \nu_l - \mu_i^\phi \log \mu_i + \mu'_i \log \mu'_i & \text{if } l \in \Gamma_S(i) \text{ and } l \neq j \\ \nu_l & \text{otherwise} \end{cases} \quad (2.56)$$

and

$$\chi'_k = \chi_k - \sum_{l \in A_{i,j}(k)} \frac{\pi(r_l)}{\omega_{\phi, l}} + \sum_{l \in A_{i,j}(k)} \frac{\pi(r_l)}{\omega'_{\phi, l}}. \quad (2.57)$$

χ_i is always recalculated statically instead

These variables can be applied to Equations (2.51) and (2.52) directly to obtain their values.

The expressions of $X'(S)$, $X'(R)$ and $X'(S, R)$ used to dynamically update Equations (2.41), (2.42) and (2.44) are

$$X'(R) = X(R) - \delta_{\omega'_j, 0} \pi(r_j) \log \pi(r_j) + \delta_{\omega_j, 0} \pi(r_j) \log \pi(r_j), \quad (2.58)$$

$$X'(S, R) = X(S, R) - \sum_{l \in \Gamma_S(i) \setminus \{j\}} x(r_l) + \sum_{l \in \Gamma_S(i) \setminus \{j\}} x'(r_l) - (1 - \delta_{\omega_j, 0}) x(r_j) + (1 - \delta_{\omega'_j, 0}) x'(r_j) \quad (2.59)$$

and

$$X'(S) = X(S) - \sum_{o \in B_{i,j}(k)} x(r_o) + \sum_{o \in B_{i,j}(k)} x'(r_k) \quad (2.60)$$

Again, the full derivation and logic behind these equations is given in Appendix B.3. It is not included here for brevity, as it is a long not immediately obvious derivation.

Extreme cases and invariants

For verification purposes (see Section 3.4), the values of the entropies along with their invariants are also given here.

Extreme cases These cases can be easily derived from Equations (2.42), (2.44) and (2.41) ($H(S, R)$, $H(S)$ and $H(R)$ respectively) by assuming the corresponding condition and applying elementary algebra.

- For a single edge, $H(S), H(R), H(S, R) = 0$.
- For a complete graph, $H(S) = \log n, H(R) = H_\pi(R), H(S, R) = H_\pi(R) + \log n$
- For a one-to-one mapping of signals into meanings with $n = m, H(S), H(R), H(S, R) = H_\pi(R)$

Invariants These invariants follow from basic information theory. [5]

- $0 \leq H(S) \leq \log n$
- $0 \leq H(R) \leq H_\pi(R)$
- $0 \leq H(S, R) \leq H_\pi(R) + \log n$

2.2 Computational aspect

This section covers the computational side of the model. This side is based on the mathematics covered in Section 2.1. Both models are seen separately, Section 2.2.1 covers the internal model while Section 2.2.2 covers the external model. Both sections cover the computational cost of computing the variables of the model, both completely (static calculation) and the change after a mutation

to $a_{i,j}$ (dynamic calculation) as well as the changes that take place by treating the case $\phi = 0$ separately.

A summary table comparing the computational costs of each of the models and the particular cases here is given at the end as Table 2.1.

2.2.1 The internal model

This section covers the computational cost of the calculation of the entropies of the internal model.

Static

A , E , μ and ω are always calculated dynamically and updated whenever an edge is added or removed to the graph, as it is very simple to do so. All entropies are calculated statically in a single loop iterating over every edge in E . The joint entropy $H(S, R)$ (Equation (2.11)) and the normalization factor M_ϕ (Equation (2.6)) are updated on every step of the loop. μ_ϕ and ω_ϕ are updated on every iteration as well. $H(S)$ (Equation (2.12)) is only updated when $\mu_{\phi,i}$ for a particular word s_i has been fully recalculated. $H(R)$ (Equation (2.13)) is updated in the same way as $H(S)$, whenever $\omega_{\phi,j}$ for a particular meaning r_j has been fully recalculated. The cost of the static calculation of entropies is $\mathcal{O}(M)$.

Dynamic

In the case of the dynamic calculation of entropies, the algorithmic cost is dominated by the computation of $X'(S)$, $X'(R)$ and $X'(S, R)$ (Equations (2.31), (2.32) and (2.33)). It can be seen immediately that looping over all the neighbours of s_i and r_j is necessary in order to update all entropies. The cost of the dynamic calculation of entropies is then $\mathcal{O}(\max(\mu_i, \omega_j))$.

The case $\phi = 0$

In order to speed up calculations, simpler equations for the case $\phi = 0$ are derived. $X'(S, R)$ (Equation (2.31)) is no longer used, as $H(S, R) = \log M$ when $\phi = 0$ (see Equation (2.11)).

$X'(S)$ (Equation (2.32)) becomes

$$X'(S) = X(S) - \mu_i \log \mu_i + \mu'_i \log \mu'_i \quad (2.61)$$

using the convention $0 \log 0 = 0$ where necessary.

Similarly to $X'(S)$, $X'(R)$ (Equation (2.33)) becomes

$$X'(R) = X(R) - \omega_j \log \omega_j + \omega'_j \log \omega'_j \quad (2.62)$$

In the case $\phi = 0$, M_ϕ (Equation (2.6)) becomes M , the number of edges in the graph.

As can be seen from equations (2.61) and (2.62), the cost of the dynamic calculation is greatly reduced when $\phi = 0$, becoming $\mathcal{O}(1)$.

2.2.2 The external model

This section overviews the computational cost of the calculation of the entropies of the external model.

Static

As with the internal model (see Section 2.2.1), A , E , μ and ω are dynamically updated whenever an edge is added or removed from the graph while all other variables, including entropies, are calculated statically.

As can be seen from Equations (2.41), (2.42) and (2.44), a loop over every edge in the graph is needed in order to recalculate all entropies and variables. Unlike the internal model, however, this loop needs to be repeated twice.

During the first run, ω_ϕ (Equation (2.4)) and ν (Equation (2.43)) are updated on every iteration. ρ (Equation (2.36)), $H(R)$ (Equation (2.41)) and $H(S, R)$ (Equation (2.42)) are calculated only on the iterations where $\omega_{\phi,j}$ and ν_j have been completely calculated for a single meaning r_j .

This leaves χ and $H(S)$ to be calculated during the second run of the loop over all edges. The computation of χ_i (Equation (2.40)) for a word s_i requires $\omega_{\phi,j}$ to be fully computed for every meaning r_j . This is the reason for running two separate loops, fully calculating ω_ϕ in one run so that χ may be calculated in the other. $H(S)$ (Equation (2.44)) is updated only on the iterations where χ_i has been completely calculated for a single word s_i .

The cost of the static calculation of entropies is the same as in the internal model, $\mathcal{O}(M)$.

Dynamic

The dynamic calculation of entropies is dominated by the computation of $X'(S)$, $X'(S, R)$ and χ (Equations (2.48), (2.49) and (2.57)). These three values iterate on three different sets: $\Gamma_S(i)$, $A_{i,j}$ and $B_{i,j}$ (Equations (2.26), (2.53) and (2.54)). It is clear from the definition of set $A_{i,j}$ is lesser or similar (one more element) in size to the set $\Gamma_S(i)$, but $\Gamma_S(i)$ will usually be bigger. The size of the set $B_{i,j}$ depends on the structure of the graph.

The cost of the dynamic calculation of entropies is then $\mathcal{O}(\max(\mu_i, |B_{i,j}|))$

The case $\phi = 0$

In order to speed up calculations, simpler equations for the case $\phi = 0$ are derived. Much of the dependency on neighboring nodes is removed.

$X(R)$ and ρ (Equations (2.58) and (2.55)) are not affected as they do not depend on ϕ . They remain simple.

$x(r_j)$ becomes

$$(1 - \delta_{w_j,0})\pi(r_j) \log \frac{\pi(r_j)}{\omega_j} \quad (2.63)$$

	Internal model	External model
Static	$\mathcal{O}(M)$	$\mathcal{O}(M)$
Dynamic ($\phi \neq 0$)	$\mathcal{O}(\max(\mu_i, \omega_j))$	$\mathcal{O}(\max(\mu_i, B_{i,j}))$
Dynamic ($\phi = 0$)	$\mathcal{O}(1)$	$\mathcal{O}(\mu_i)$

Table 2.1: Summary table of the computational cost of each model and for particular cases.

and so no longer depends on neighbors of r_j . Consequently, $X(S, R)$ also no longer depends on a set of neighbors and becomes

$$X'(S, R) = X(S, R) - x(r_j) + (1 - \delta_{\omega'_j, 0})x'(r_j). \quad (2.64)$$

χ_i (Equation (2.40)) is simplified but still depends on the neighborhood of s_i in order to be updated

$$\chi'_k = \begin{cases} \chi_i + (1 - a_{i,j}) \frac{\pi(r_j)}{\omega'_j} - a_{i,j} \frac{\pi(r_j)}{\omega_j} & \text{if } k = 1 \\ \chi_k - \frac{\pi(r_j)}{\omega_j} + \frac{\pi(r_j)}{\omega'_j} & \text{if } k \in \Gamma_R(j) \text{ and } k \neq i \\ \chi_k & \text{otherwise.} \end{cases} \quad (2.65)$$

As $X(S)$ depends on χ (Equation (2.49)), $X'(S)$ becomes

$$X'(S) = X(S) - \sum_{k \in \Gamma_R(j) \cup \{i\}} x(s_i) + \sum_{k \in \Gamma_R(j) \cup \{i\}} x'(s_i) \quad (2.66)$$

with (see Equation (2.51))

$$x(s_i) = \chi_i \log \chi_i. \quad (2.67)$$

$X(S)$ and χ need to iterate the neighbors of s_i , and so the computational complexity is $\mathcal{O}(\mu_i)$.

Chapter 3

Methods

This chapter centers on the implementation of the models defined in Chapter 2 and the optimization method outlined in Section 1.3.6. Section 3.1 focuses on the implementation of the models in a C++ program, including a class hierarchy diagram, several important algorithms, various methods used to avoid floating point error and the different distributions of π (see Section 2.1.3) that were implemented. Section 3.2 deals with the optimization algorithm, how it changes for the static or dynamic cases and several optimizations that allow for faster runtime and more precise results. On Section 3.3 the approach taken to parallelization is outlined and justified. Section 3.4 deals with the verification of the code. Finally, Section 3.5 explains the main challenges related to numerical precision while Section 3.6 covers other problems related to the implementation.

3.1 Model implementation

This section covers the implementation of the two models seen in Chapter 2. It starts with a class hierarchy diagram of the organization of the static and dynamic versions of the two models. Following this, the algorithm used to randomly generate a random bipartite graph (given a vertex probability or given the number of edges) is outlined. Following this, the various “tricks” used in order to minimize the floating point error, specially in the dynamic version of the model are explained. To close this section, details on the various distributions of π that were implemented are given.

Class hierarchy diagram

The diagram in Figure 3.1 shows the classes implementing the bipartite graph models (see Chapter 2).

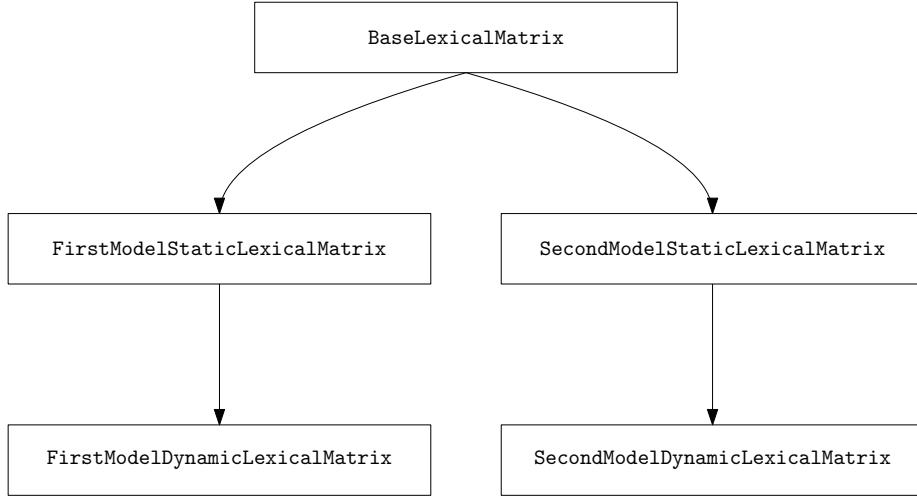


Figure 3.1: Diagram showing the hierarchy of the classes implementing the two models in their dynamic and static variants in the C++ program.

Random graph generation

This section outlines the algorithm implemented to generate a random Erdos-Renyi bipartite graph [6] in pseudocode. The graph can be generated by specifying either the probability of edge creation ($G_{n,m,p}$) as seen in Algorithm 1 or the total number of edges ($G_{n,m,e}$) as in Algorithm 2. Note that all Algorithm 1 actually does is compute a value of e then call Algorithm 2.

citar alguna font? explicar d'on surt la notació $G_{n,m,e}$

Algorithm 1 $G_{n,m,p}$ algorithm to generate a random $n \times m$ graph given probability of edge p .

```

1: procedure  $G(n, m, p)$ 
2:    $e \leftarrow \text{BINOM}(n \cdot m, p)$ 
3:   return  $G(n, m, e)$ 
4: end procedure

```

Dealing with floating point error

One of the major challenges (see Section 3.5) has been dealing with floating point error. Two major “tricks” have been used in order to minimize this loss of precision as much as possible.

The Accumulator class Whenever possible, specially in the static calculations, additions have been done through an accumulator class. This class main-

Algorithm 2 $G_{n,m,e}$ algorithm to generate a random $n \times m$ graph given the number of edges e .

```
1: procedure  $G(n, m, e)$ 
2:   if disconnected meanings disallowed then
3:     for each  $r_j$  such that  $1 \leq j \leq m$  do
4:        $s_i \leftarrow \text{UNIFORM}(s_1, s_n)$ 
5:       ADD EDGE( $s_i, r_j$ )
6:        $e \leftarrow e - 1$ 
7:     end for
8:   end if
9:   while  $e > 0$  do
10:     $s_i \leftarrow \text{UNIFORM}(s_1, s_n)$ 
11:     $r_j \leftarrow \text{UNIFORM}(r_1, r_n)$ 
12:    ADD EDGE( $s_i, r_j$ )
13:     $e \leftarrow e - 1$ 
14:  end while
15: end procedure
```

tains two variables, `total` and `current`. When a value is added, it is added to the `current` variable. If the `current` variable is greater than `total`, it is added to `total` and reset to 0. This is an inexpensive and effective way to minimize the loss of precision from a common occurrence during static calculation, the addition of a smaller value into a greater one.

Static recalculation During the optimization process, whenever a new minimum is reached, a static recalculation is forced. In this way, one of the major source of floating point error from the dynamic version is removed. The dynamic equations (see Section 2.1) consist of many additions and subtractions. These additions and subtractions are a major contributor to the accumulation of floating point error. Since finding a new minimum is not common, this does not affect the run time of the optimization process.

Save and restore Most steps of the optimization process will not improve the cost function (see Section 3.2). Instead of undoing the step and recalculating all the parameters of the model, the parameters are saved before performing the step, and they are restored when the step needs to be reverted. This strategy reduces the amount of additions and subtractions to be done when the recalculation of parameters is dynamic, thus reducing the amount of floating point error.

Distribution of π

The second model (see Section 2.1.3) is based on a vector of length m of *a priori* probabilities π . Uniform, geometric, power law and broken stick probability

distributions have been implemented in order to give values to the elements of this vector.

Uniform distribution The uniform distribution simply assigns the probability $1/m$ to every element in the vector.

Geometric distribution Assigns a right truncated geometric distribution with parameter p to the elements of π , following

$$(1 - p)^{x-1} \frac{p}{1 - (1 - p)^m}.$$

This formula can be easily derived from the more general formula for a truncated discrete probability distribution

$$P(X = x | a < x \leq b) = p(X = x) / (P(X = b) - P(X = a)).$$

Power law distribution Assigns a right truncated power law distribution with parameter α to the elements of π . It follows

$$x^{-\alpha} \sum_{i=1}^m i^{-\alpha}$$

Broken stick distribution A broken stick distribution corresponds to the distribution of the sizes of the pieces of a stick that is broken by a point along its length randomly a number $m - 1$ of times. This distribution corresponds to the expected value of the lengths of the pieces of a stick of total length 1. The formula and its derivation is given in [21] (Equation 1).

3.2 Optimization

The optimization algorithm follows the Markov chain Monte Carlo method at zero temperature. The model is initialized in some way (depending on parameters), and on each iteration a number of random mutations (adding or removing an edge on the underlying bipartite graph) are effectuated. The cost function is evaluated with a parameter λ . If it is found to have improved, the model is recalculated statically (to avoid floating point error, see Section 3.1). If it is found to not have improved, the changes are reverted. The optimization stops after a number of failures to improve the cost function. The optimization stops early if the global minimum for the cost function is reached.

The parameter λ is given as a parameter. It controls the weight of the two forces that contribute to the cost function. See Section 1.3.6.

The number of mutations done on each step depends on the parameters and may be either constant or follow a binomial distribution. In the case of a binomial distribution, it is possible that zero mutations may be effectuated. In

this case, nothing is done and the step is considered to not have improved the cost function.

The number of failures to improve the cost function needed to stop iterating depends on parameters. Two possible values are calculated from parameters of the model:

- **weak:** Enough steps are performed such that it is likely that all edges have been visited once. This follows the Coupon Collector's Problem [15], and the formula for the number of attempts is

$$\lfloor nm \log nm \rfloor$$

- **strong:** Enough steps are performed such that it is likely that every possible pair of edges has been visited once. This also follows the Coupon Collector's Problem,

$$\left\lfloor \binom{nm}{2} \log \binom{nm}{2} \right\rfloor$$

3.3 Parallelization

During the initial phases of the design of the software, it was considered to add parallelization in order to speed up calculations. Parallelization would add overhead and complexity to an already complex codebase in exchange for sharing the computation between the machine's processors.

A typical task for the program is the computation of a range of values of λ (see Section 3.2) and to calculate several samples for each λ . This layout lends itself well to multiprocessing, without the program implementing any sort of parallelization scheme.

The user is expected to divide the load his or herself, for instance by generating many separate configuration files (or using a script to do so) and using a tool such as one of the many implementations of `parallel` or a computing cluster's workload manager. In this way, the operating system handles the parallelization complexity and the software's complexity and overhead is reduced. This also makes verification much easier.

3.4 Verification

Verification is a vital part of any software development process. In this case, verification involves ensuring that both static and dynamic versions of the model are correctly implemented. Dynamic versions are specially complex and so need to be tested thoroughly.

The test routines implemented verify that the models give “sane” results by checking extreme cases (see Sections). In addition, the invariants of the entropies (also in Sections) are verified after every mutation and the program is aborted if they ever do not hold. A test also checks the invariants exhaustively

for every possible combination of connected and disconnected edges for very small graphs ($n = m = 4$).

Other functionality less related with the implementation of the model is also tested.

- The save/restore functionality described in Section 3.1 for dealing with floating point error.
- Quickly testing whether the matrix has disconnected meanings (configuration parameter disallows disconnected meanings).

The last test is running the program for relatively small values of n and m and all combinations of parameters, and exhaustively checking that both static and dynamic versions give identical output for the same seed. This test is used to ensure the correctness of the dynamic version.

3.5 Numerical precision problems

Challenges and problems related to the numerical precision are given in this section.

In both the static and dynamic versions of the calculations, there are many additions or subtractions of floating point numbers. These actions often involve a loss of precision due to the difference in magnitude of the values being operated. Over many iterations, this loss of precision can become apparent and change the final result of the simulation. See Section 3.1 where an entire subsection is dedicated to dealing with numerical error.

The geometric distribution decreases in probability very quickly. Because of this, erroneous values can be produced when the value of π is too close to 0. As outlined in Section 2.1.3, the values of π should never be zero. This problem occurs in general whenever π is too close to zero and the program automatically warns about it if the parameters will generate values of π that are too small. For this reason, the geometric distribution does not appear in Chapter 4

3.6 Other problems

Since the runtime of the dynamic versions often depends on the number of neighbors of the affected vertices (see Section 2.2), running calculations on complete or almost complete graphs takes a long time, as the advantages of the dynamic formulas are reduced significantly. As a result of this, running the optimization with an initial complete graph can take an unacceptable amount of time when compared to other initial conditions.

The strong stop condition (Section 3.2) results far too many attempts to improve the cost function when $n = m$ is large (for $n = m = 400$, the number of attempts is of the order of 10^{12}). For this reason, the strong stop condition is not used in Chapter 4

Chapter 4

Results

In this chapter the results generated with the open source tool are presented. This chapter is divided into two sections. Section 4.1 presents results from previous papers recreated using the tool. Section 4.2 presents results from the newer family of models introduced in [13]

4.1 Verification of previous results

In this section a replication of previous results is presented. This allows to verify the model and its implementation against previous experiments.

However, in some cases there was limited information in the original papers about the parameter used in the experiments. In other cases, there were errors in previous papers. In both cases, determining the correct parameter becomes a matter of trial and error.



The ϕ parameter was not considered in either of the papers replicated in this section, and so it is set to 0.

Section 4.1.1 replicates the results from [9], which correspond to the internal model from this thesis. Section 4.1.2 replicates the results from [12], corresponding to the external model.

4.1.1 Results from the internal model (2005)

Figure 4.1 shows both the original Figure 2 from [9] and a recreation of this figure using the new tool. Figure 4.2 shows both the original Figure 3 from [9] and a recreation of this figure, also generated using the new tool.

The initial graph was not specified in [9] and a $G_{n,m,1/n}$ ($n = m$) graph is used in the replication. The paper also does not specify how to stop the optimization process and so the weak stop condition (Section 3.2) is used.

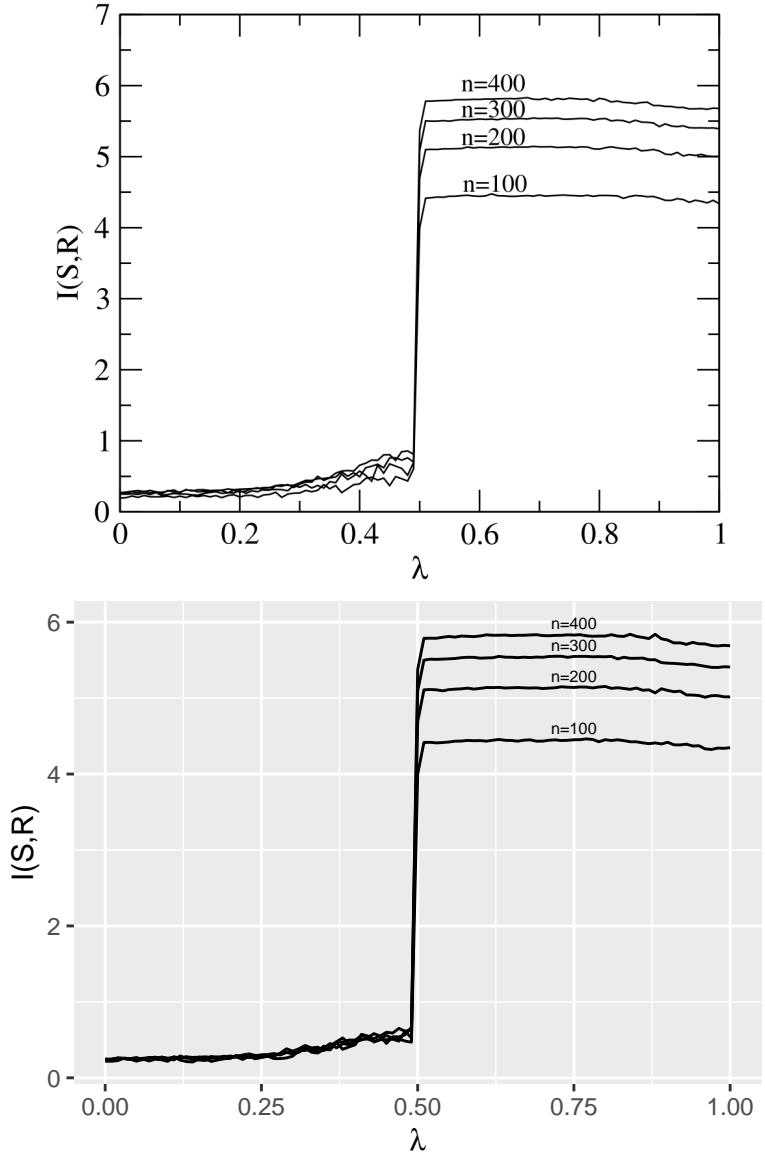


Figure 4.1: Above: Figure 2 from [9].

Below: Recreation of that same figure using the open source tool created for this thesis.

 The mutual information $I(S, R)$ is in the y axis and the λ parameter on the x axis. Graphs of different sizes are shown, $n = m = 100$, $n = m = 200$, $n = m = 300$, $n = m = 400$. Averages over 30 realizations. $\phi = 0$, the initial graph is $G_{n,m,1/n}$, each iteration of the optimization algorithm performs 2 mutations on the graph, the weak stop condition is used to stop the optimization process, unlinked objects are allowed.

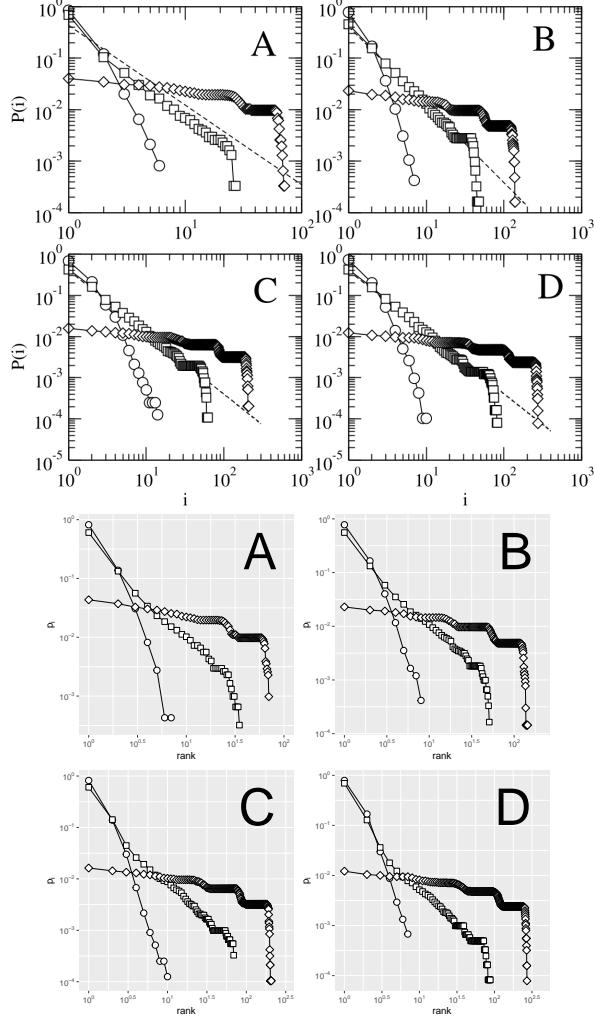


Figure 4.2: Above: Figure 3 from [9].

Below: Recreation of that same figure using the open source tool created for this thesis.

 $P(i)$, the probability of the i th most frequent signal, obtained from minimum energy configurations for systems of sizes $n = m = 100$ (A), $n = m = 200$ (B), $n = m = 300$ (C), $n = m = 400$ (D). Four series are shown in each plot, $\lambda = 0.49$ (circles), $\lambda = \lambda^*$ (squares) and $\lambda = 0.5$ (diamonds) and the ideal curve for α^* . Averages over 30 realizations. When $\lambda = \lambda^*$, $\alpha^* = 1.54$ for $n = m = 100$, $\alpha^* = 1.51$ for $n = m = 200$, $\alpha^* = 1.5$ for $n = m = 300$ and $\alpha^* = 1.49$ for $n = m = 400$. It was chosen that $\lambda^* = 0.4986$ for $n = m = 100$, $\lambda^* = 0.4987$ for $n = m = 200$, $\lambda^* = 0.4987$ for $n = m = 300$, and $\lambda^* = 0.4986$ for $n = m = 400$. Averages over 30 realizations. $\phi = 0$, the initial graph is $G_{n,m,1/n}$, each iteration of the optimization algorithm performs 2 mutations on the graph, the weak stop condition is used to stop the optimization process, unlinked objects are allowed.

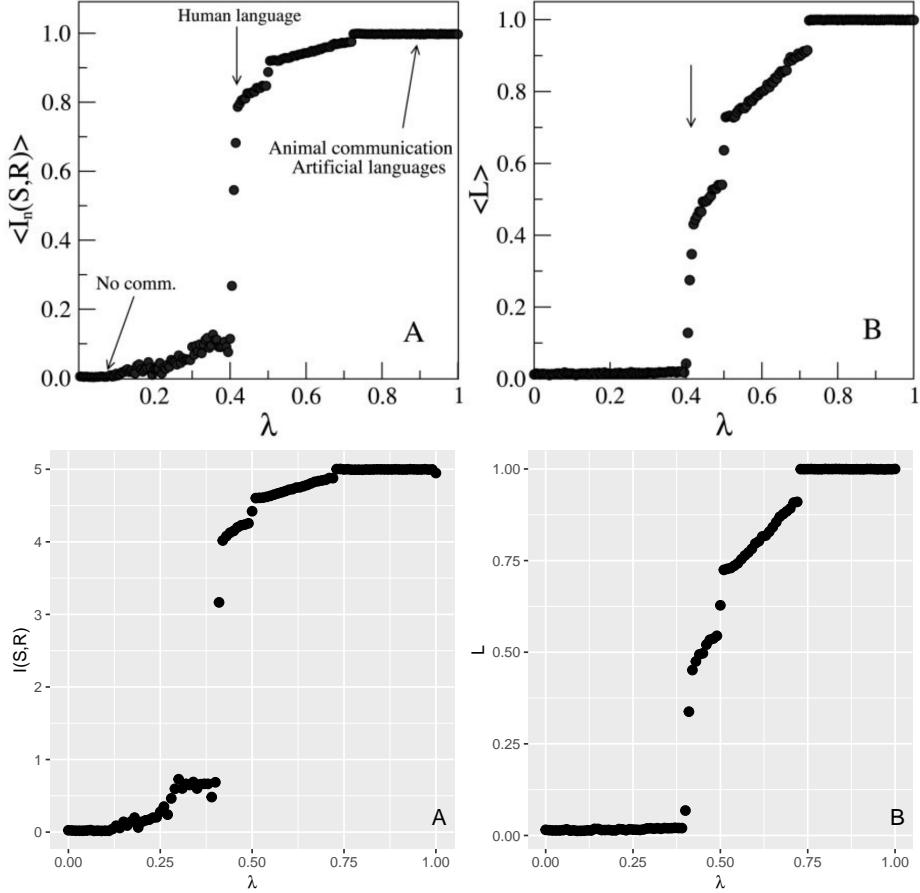


Figure 4.3: Above: Figure 2 from [12]

Below: Recreation of that same figure using the open source tool created for this thesis.

In A, $I(S, R)$ is the mutual information obtained for values of λ between 0 and 1. In B, L is the lexicon size obtained for values of λ between 0 and 1. An abrupt change is seen for $\lambda \approx 0.41$ in both A and B. Averages over 30 replicas. $n = m = 150$, $\phi = 0$, unlinked objects are not allowed, π follows a uniform distribution,

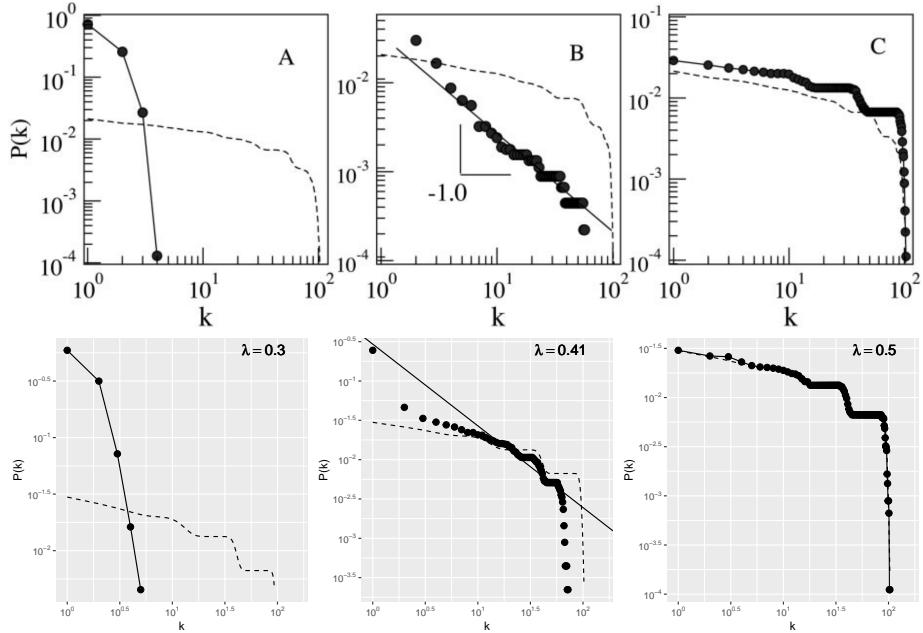


Figure 4.4: Above: Figure 3 from [12]

Below: Recreation of that same figure using the open source tool created for this thesis.

Normalized signal frequency $P(k)$ versus rank k . The dashed lines show the distribution obtained when from a G_{n,m,e^*} graph where e^* is the number of connections in these optimal configurations. In both cases the distribution of B is still consistent with human language, $\alpha = 1$. Averages over 30 replicas, $n = m = 150$, the initial graph is $G_{n,m,10/n}$, the number of mutations on each iteration follows a binomial distribution with average 1.75, the optimization stops after $2 * n * m$ mutations that do not improve the cost function.

4.1.2 Results from the external model (2003)

Figure 4.3 shows both the original Figure 2 from [12] and the recreation of this figure using the new tool. Figure 4.4 shows the original Figure3 from [12] and a recreation using the new tool.

The initial graph was specified as a $G_{n,m,\rho}$ but the value of ρ was not given in the paper, a $G_{n,m,10/n}$ graph was used (that is, a value of $\rho = \frac{10}{n}$). It is specified that the number of mutations done on each iteration of the optimization algorithm follows a binomial distribution. However it was later found that, due to a bug in the generator of binomial numbers, the number of mutations must have been lower. Several values were tried for the average number of binomial mutations, with 1.75 giving the best results.



4.2 Results for current models

4.2.1 Information Theoretic

first model, $\phi = 0$

- initial: random 4.5
- initial: single 4.6
- initial: one-to-one 4.7

first model, $\phi = 1$

- initial: random 4.8
- initial: single 4.9
- initial: one-to-one 4.10

second model, $\phi = 0$

all π uniform

- allow disconnect meanings
 - initial: random 4.11
 - initial: single 4.12
 - initial: one-to-one 4.13
- do not allow disconnect meanings
 - initial: random 4.14
 - initial: single 4.15
 - initial: one-to-one 4.16

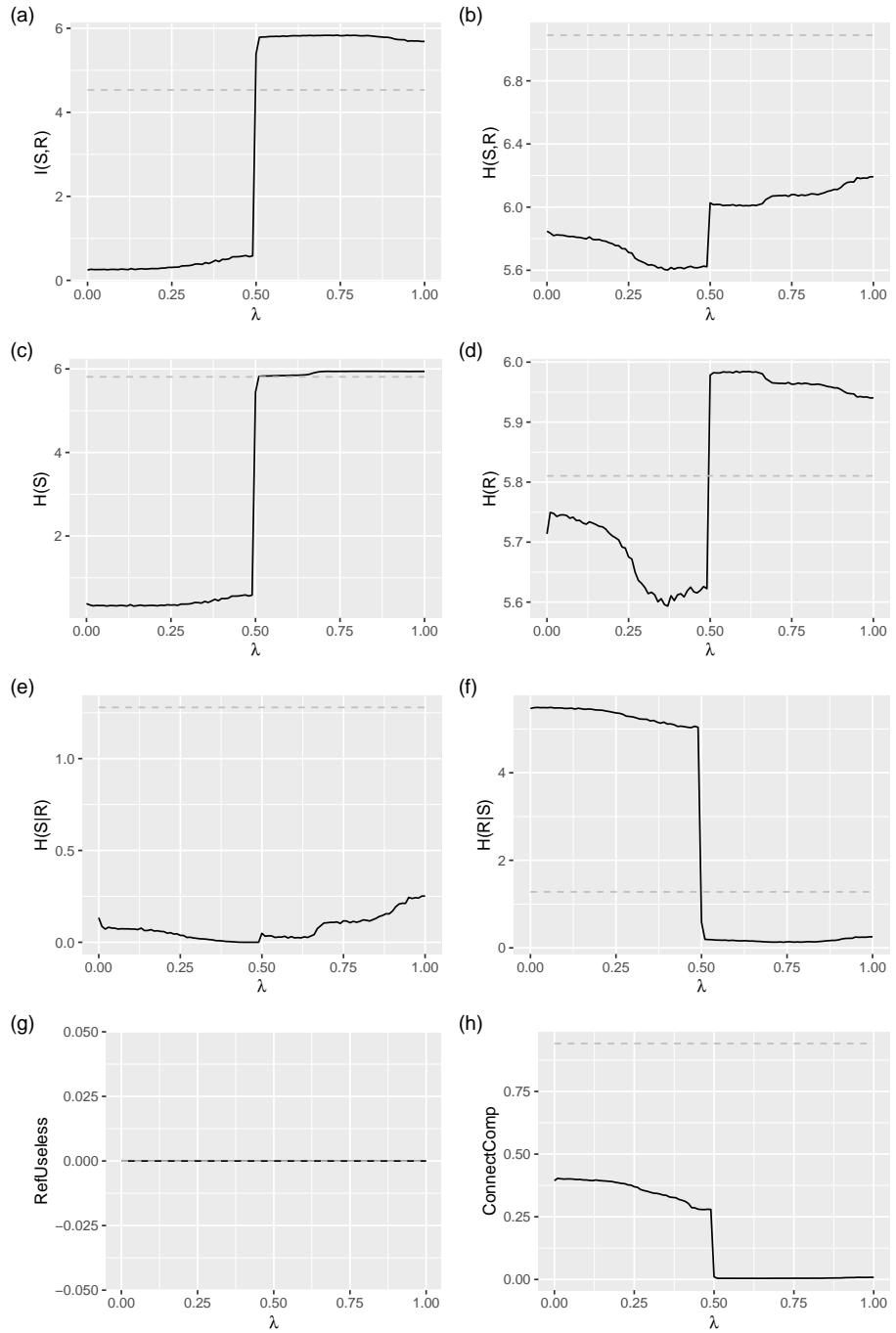


Figure 4.5: a

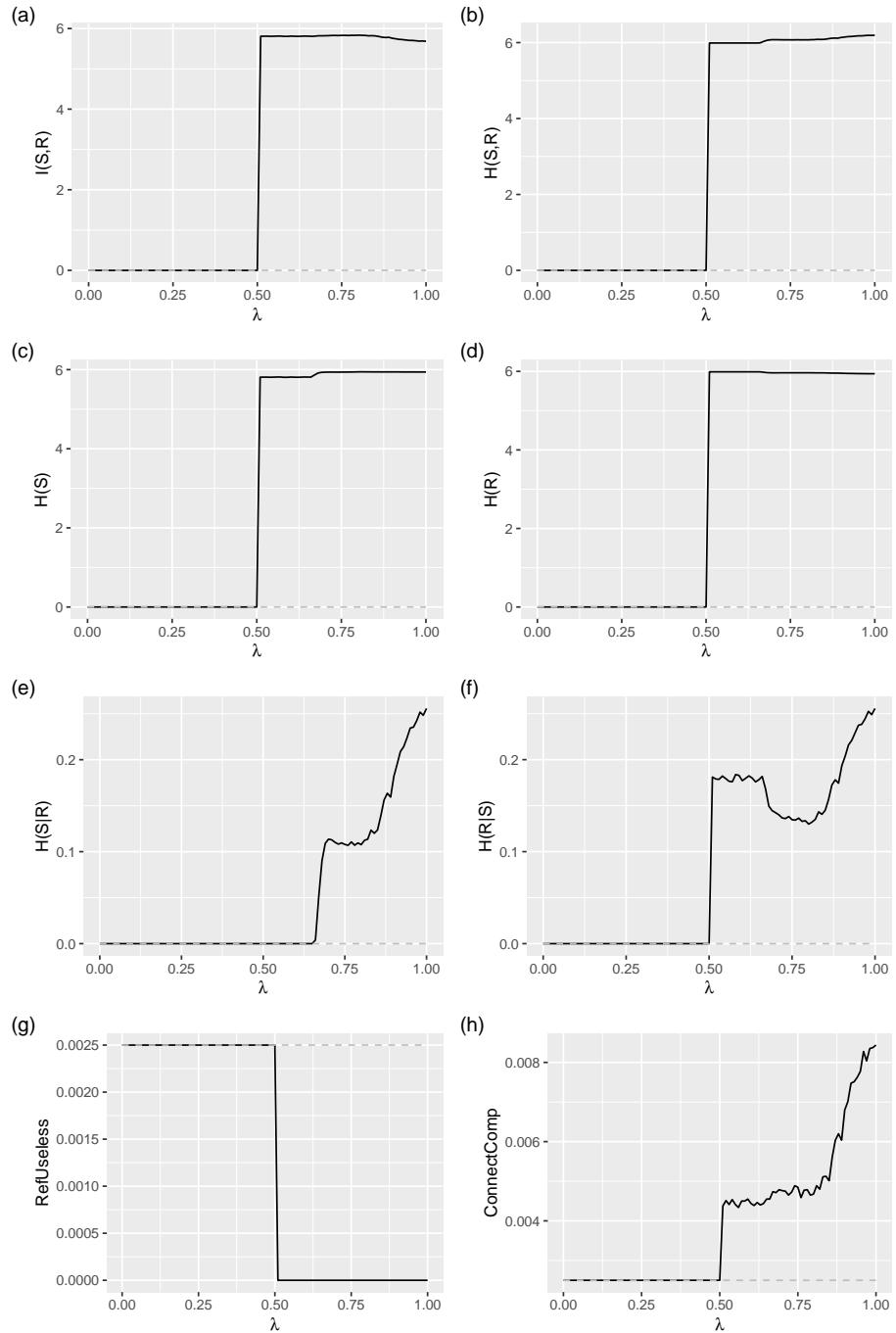


Figure 4.6: a

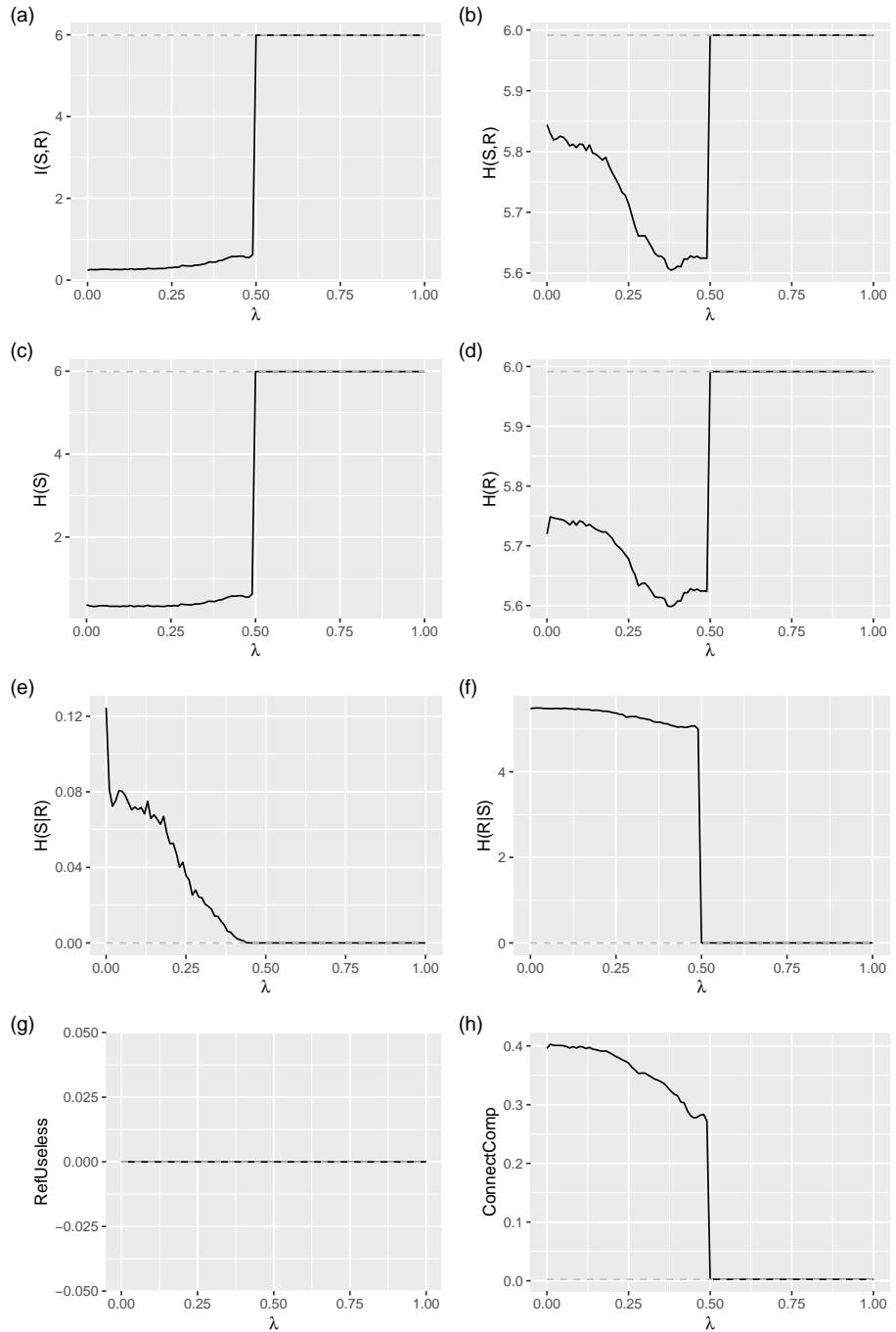


Figure 4.7: a

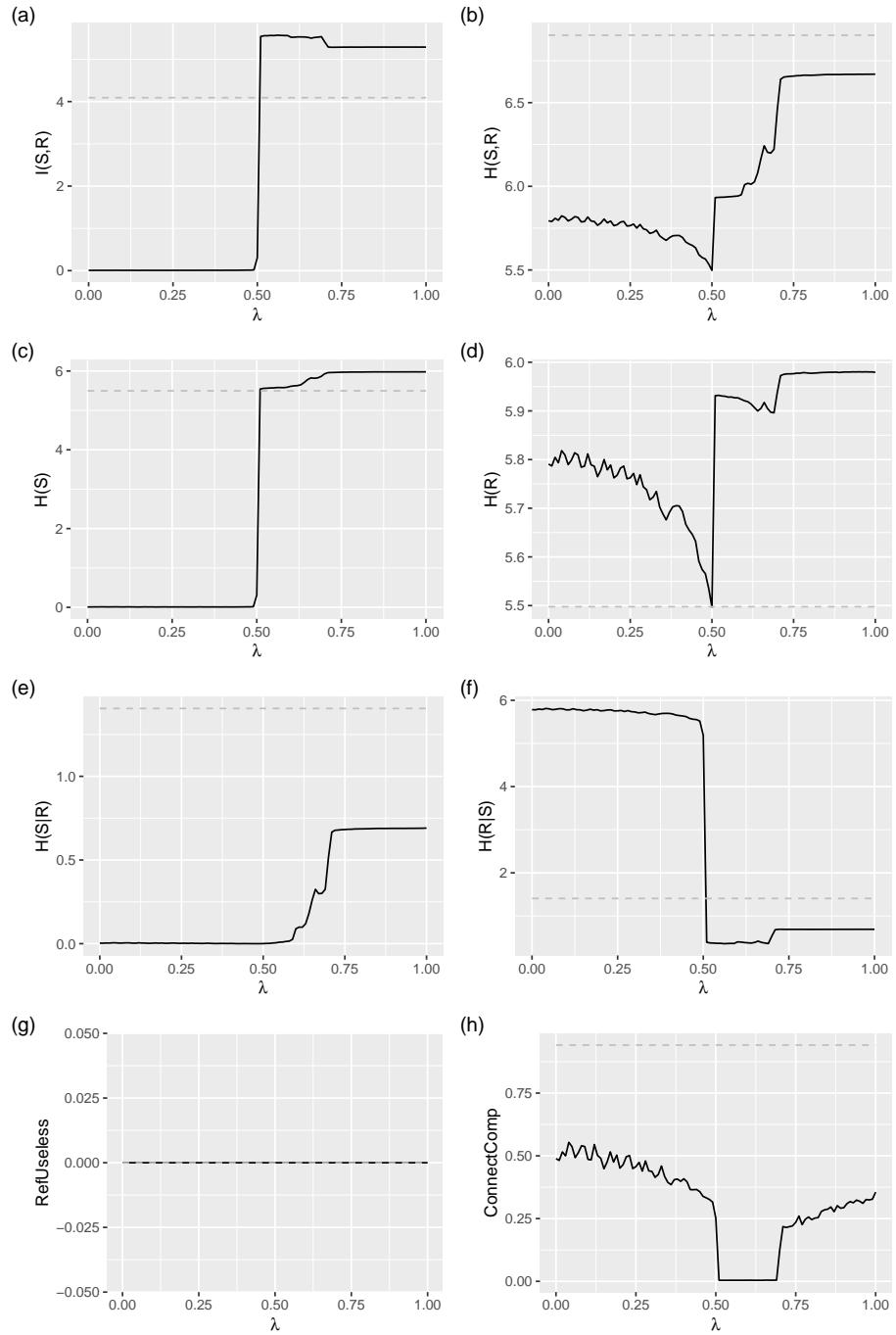


Figure 4.8: a

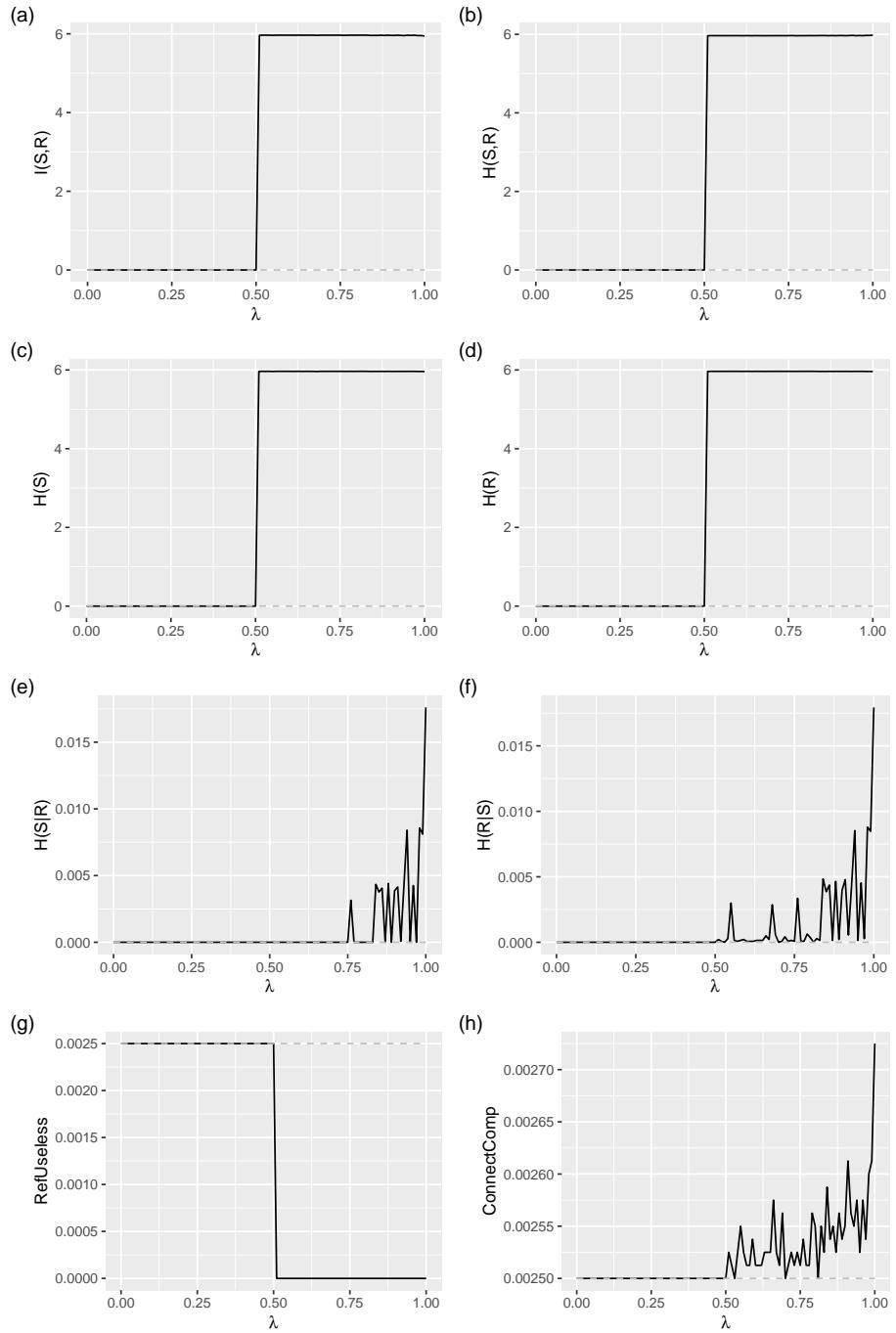


Figure 4.9: a

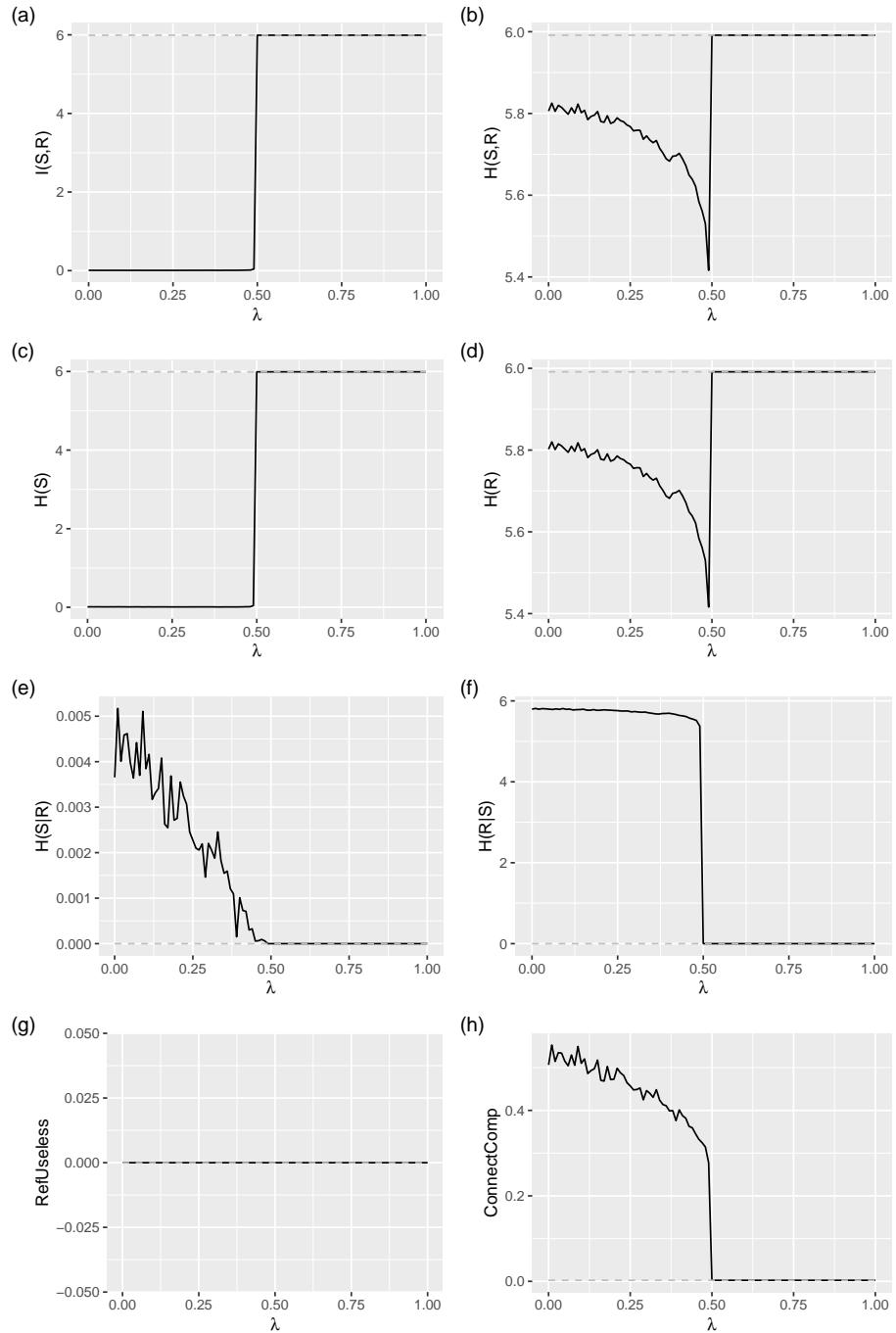


Figure 4.10: a

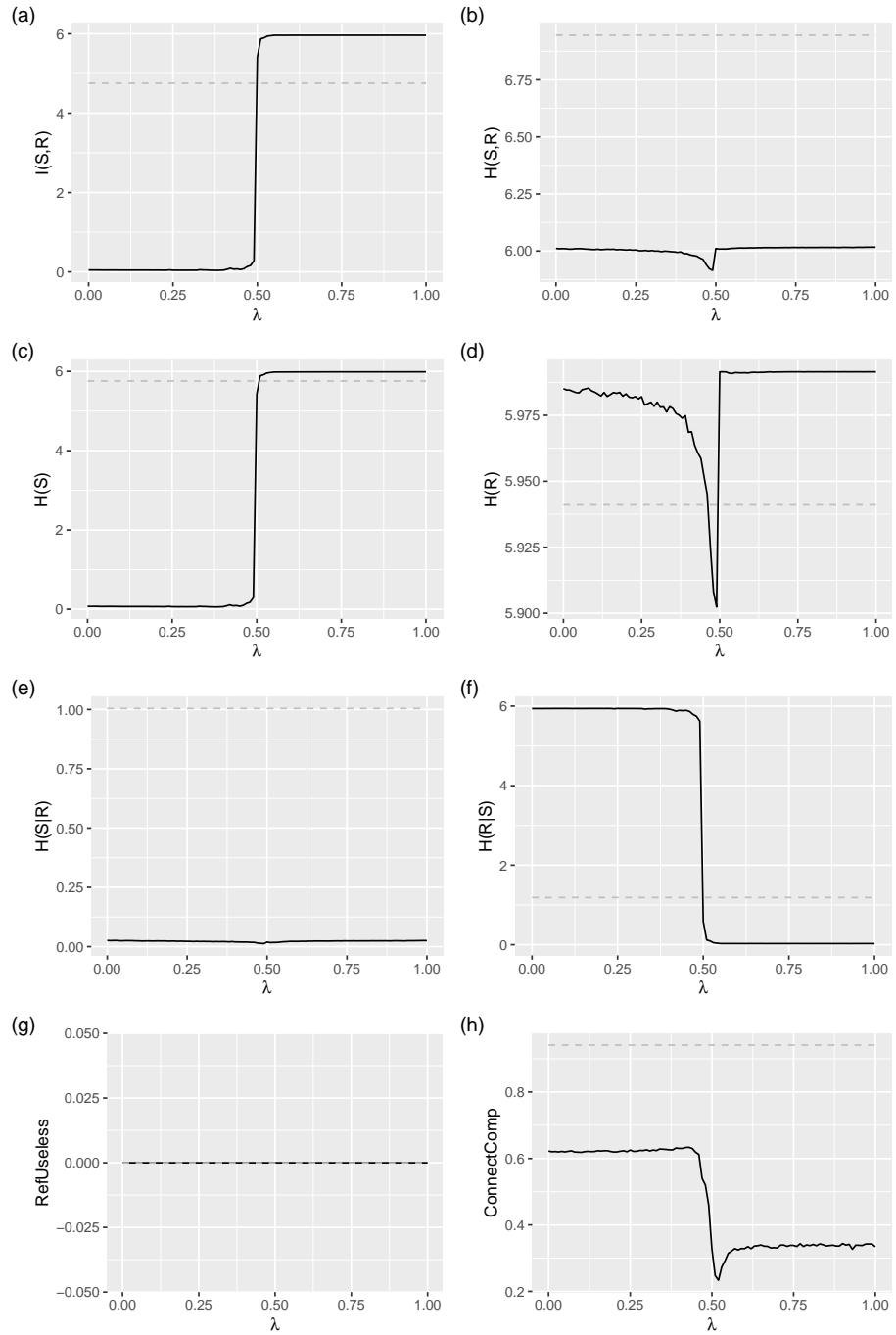


Figure 4.11: a

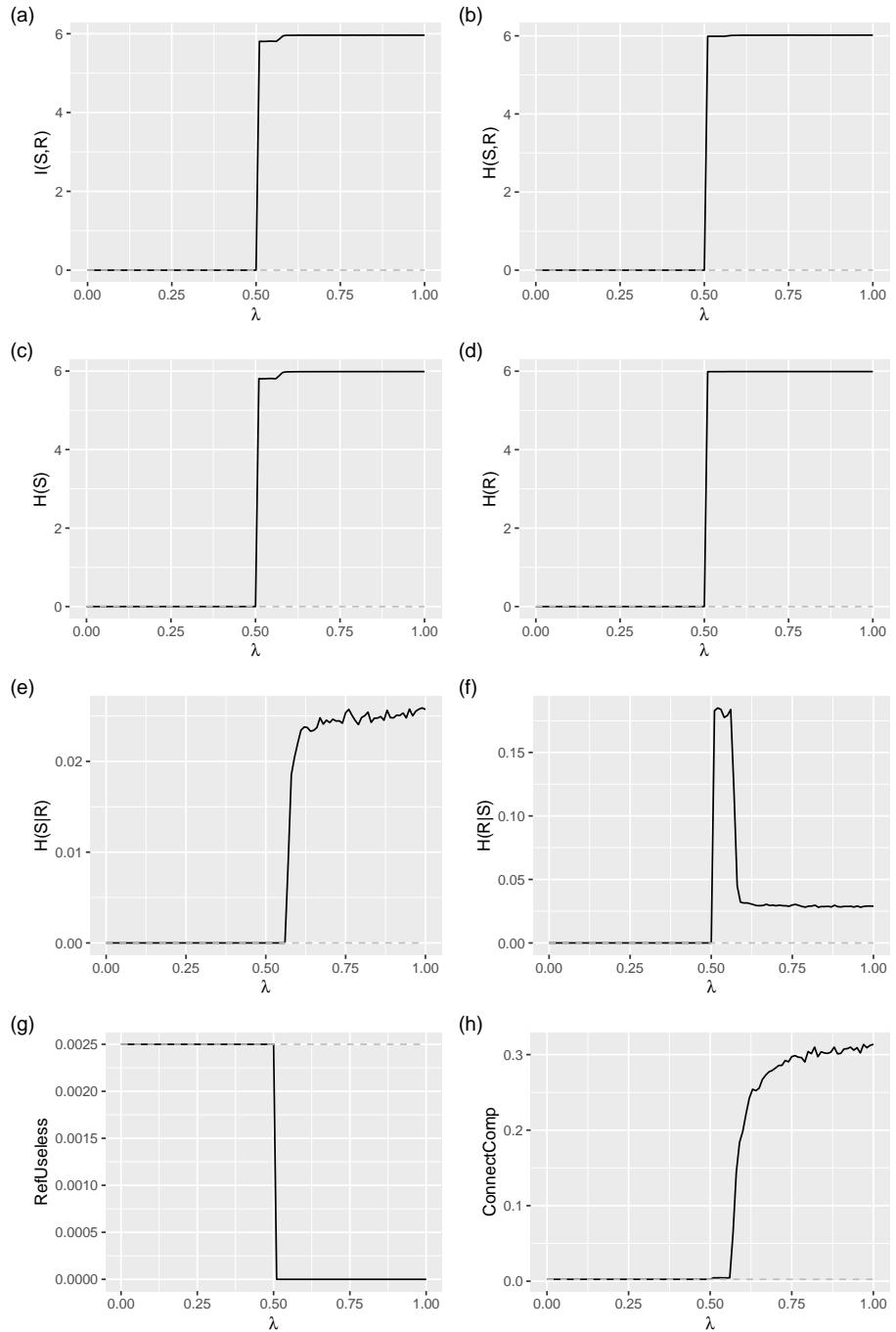


Figure 4.12: a

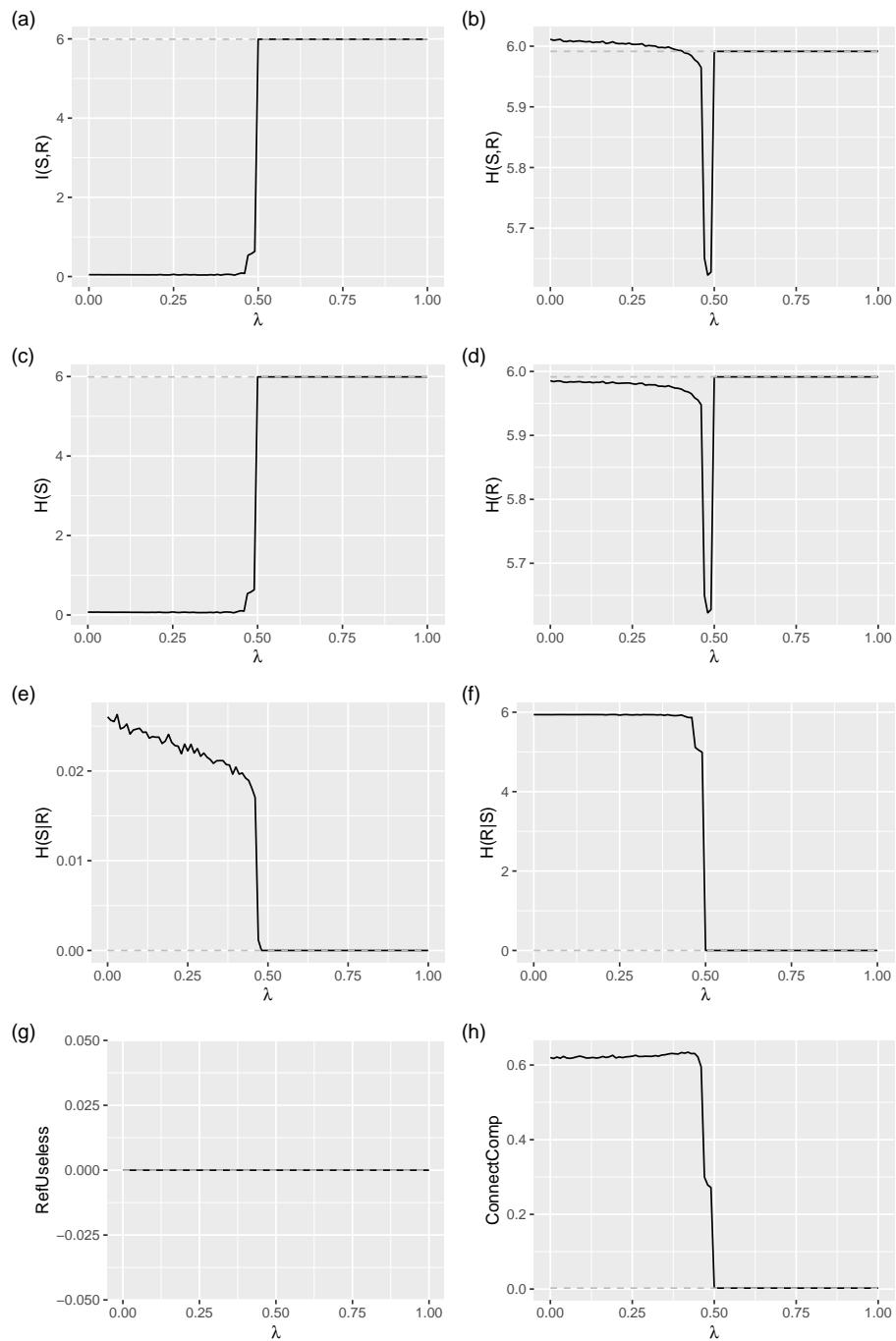


Figure 4.13: a

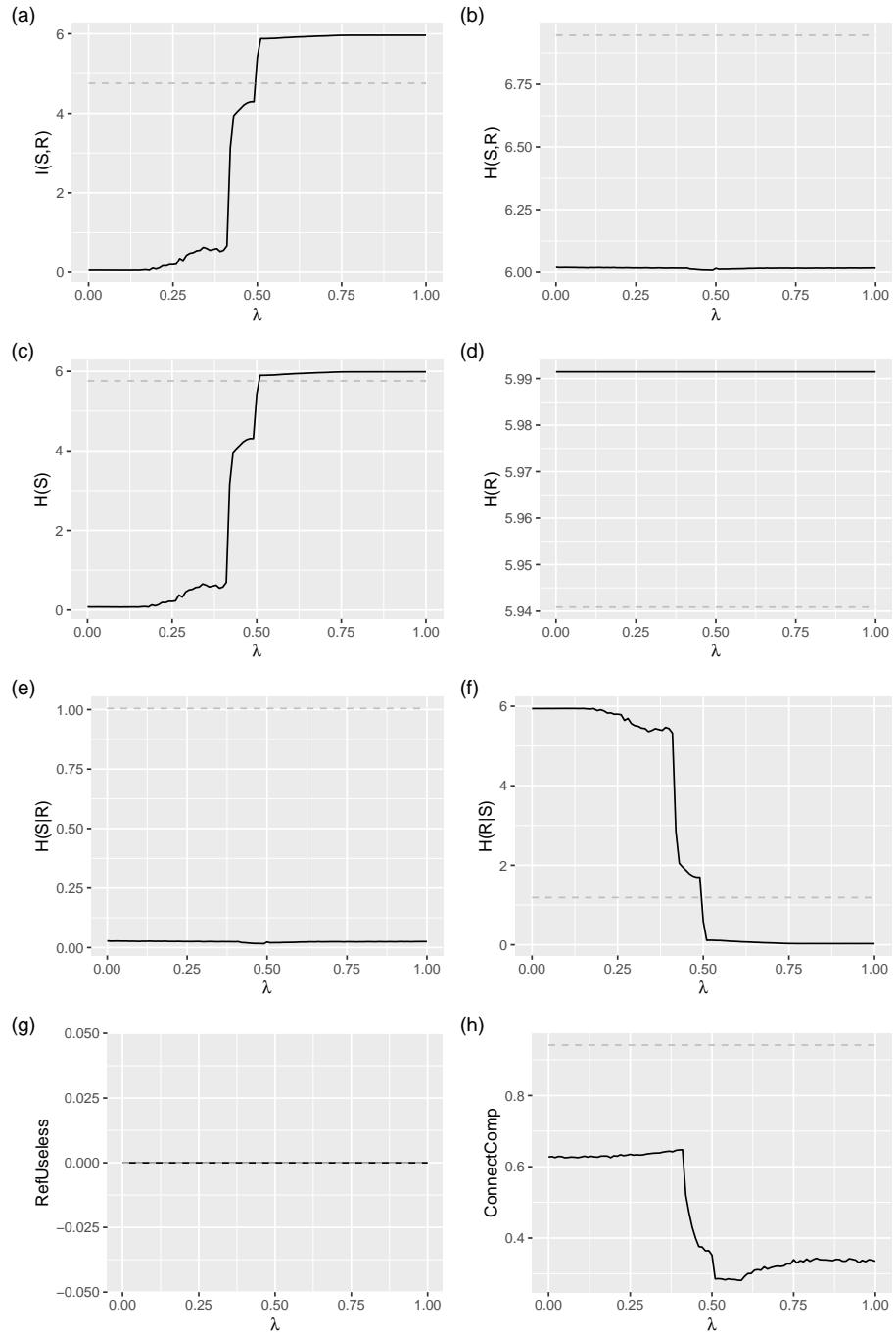


Figure 4.14: a

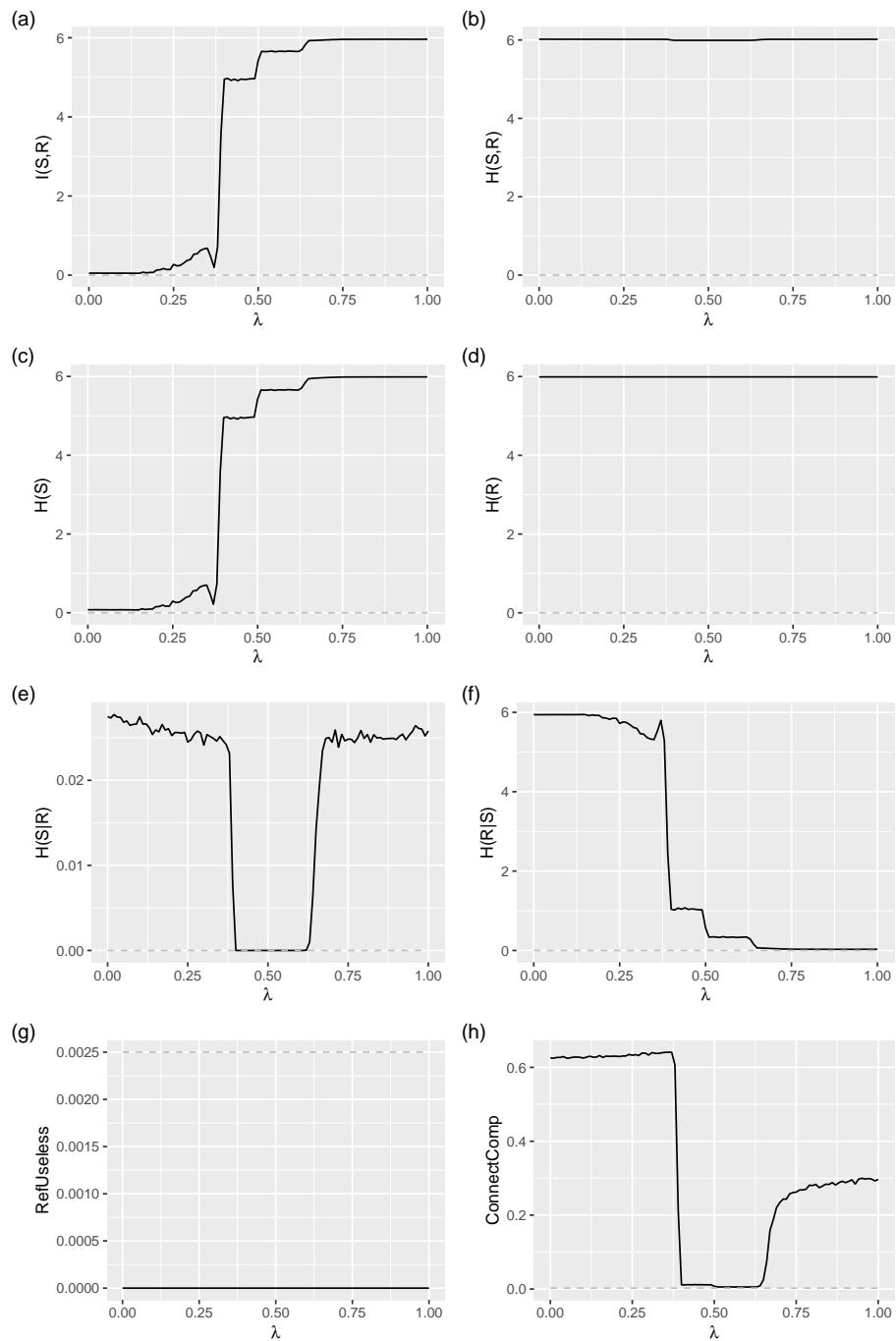


Figure 4.15: a

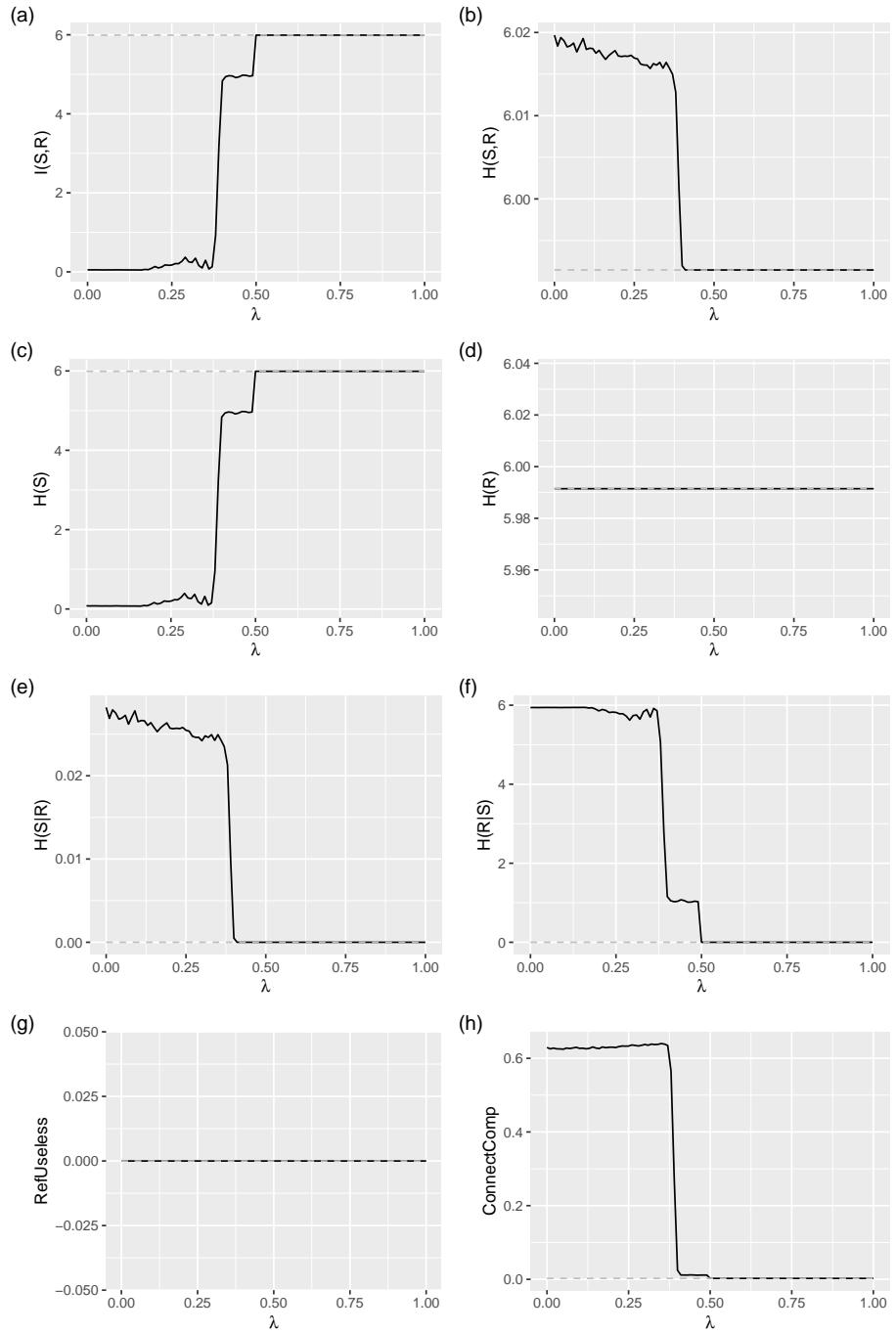


Figure 4.16: a

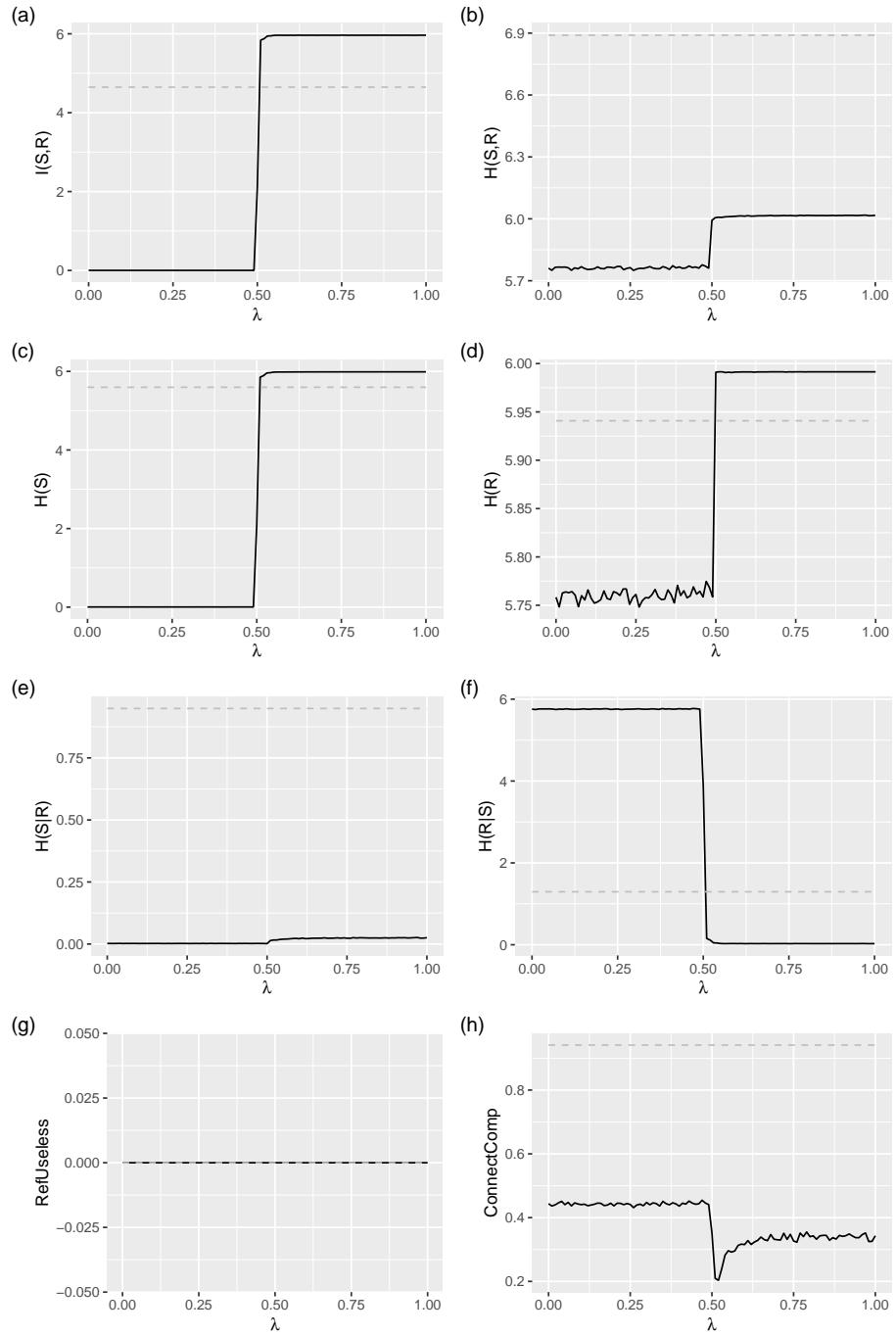


Figure 4.17: a

informationTheoretic_uniform_phi1_nm400_dynamic_singleLink_allowUnlinked.pdf

Figure 4.18: a

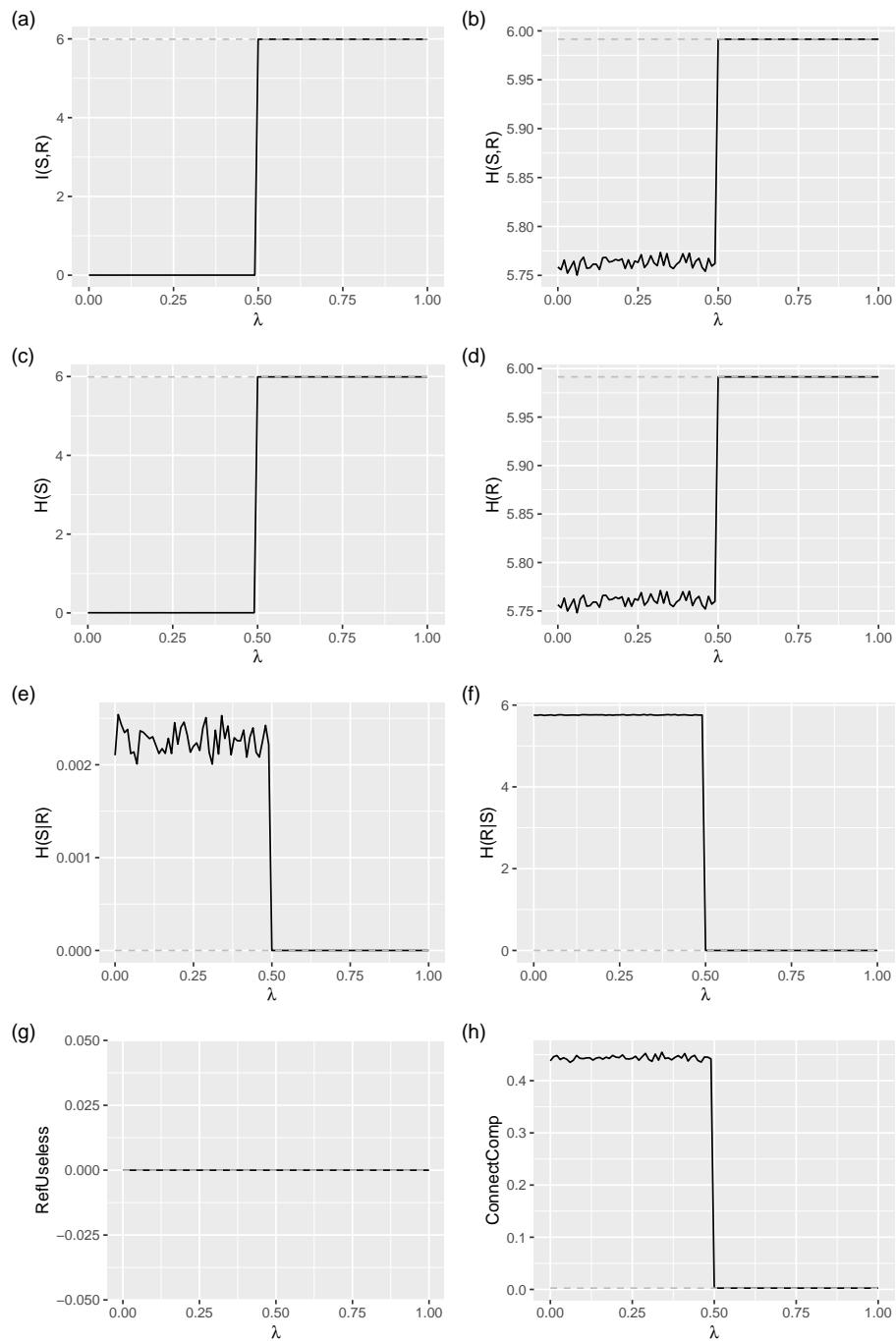


Figure 4.19: a

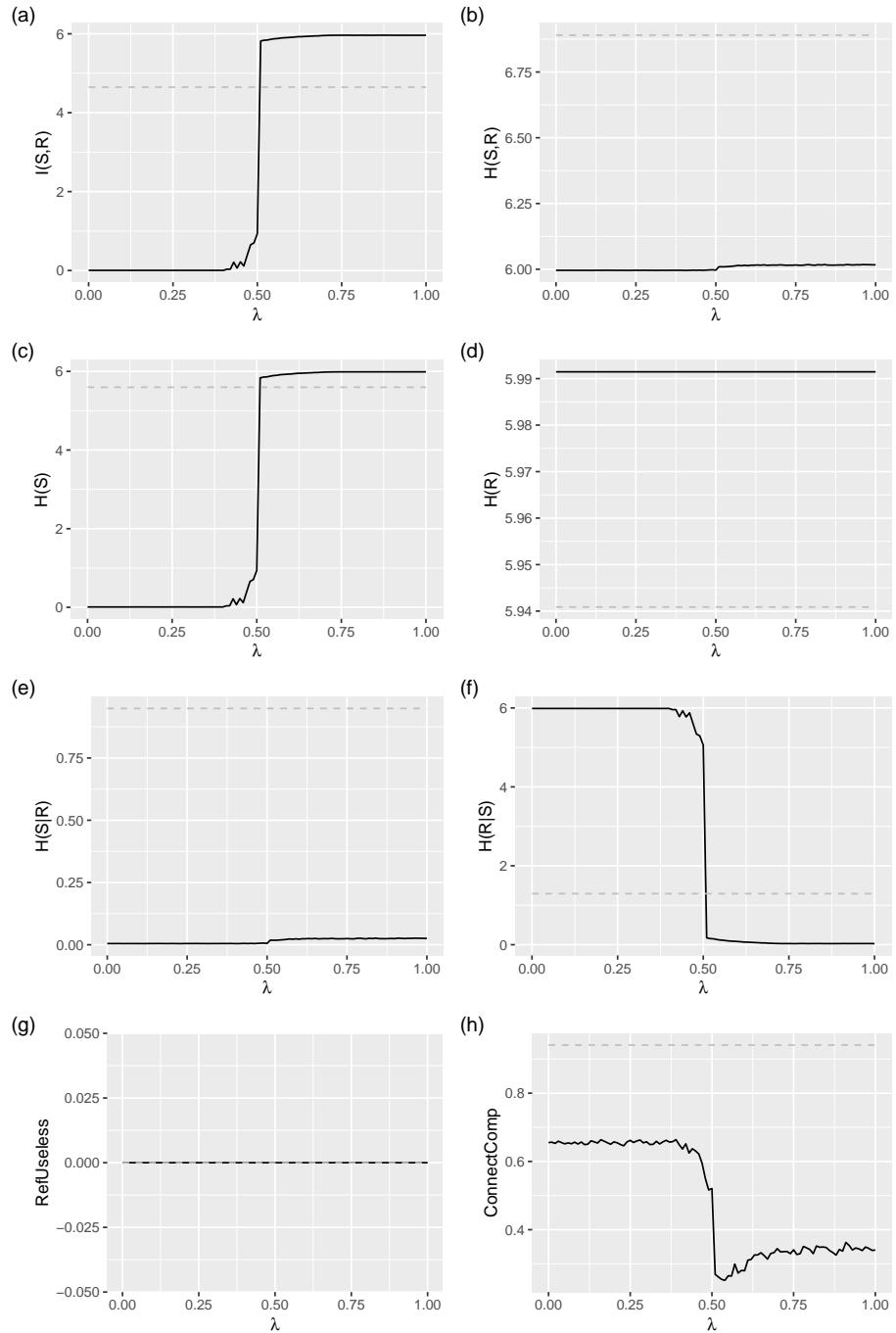


Figure 4.20: a

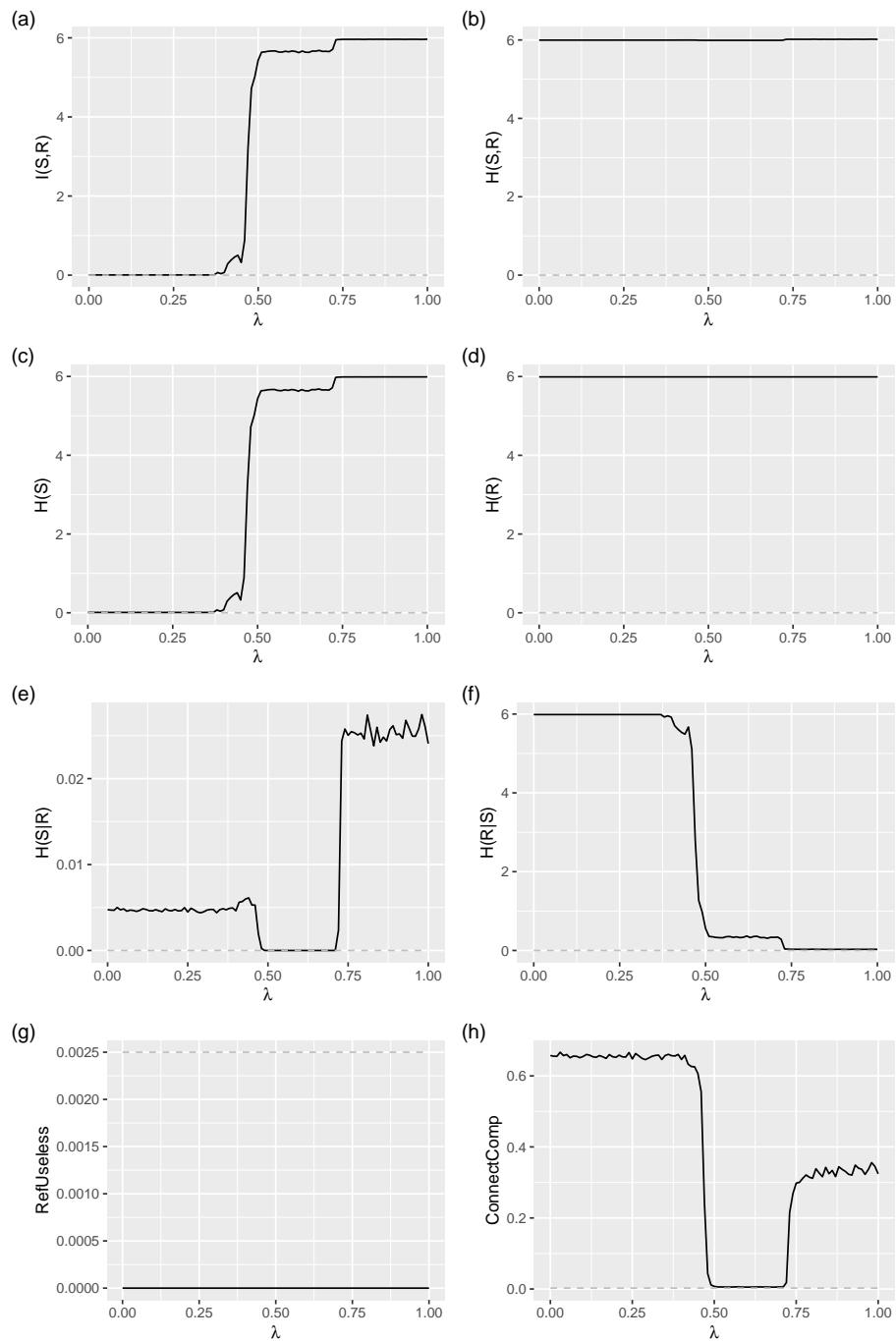


Figure 4.21: a

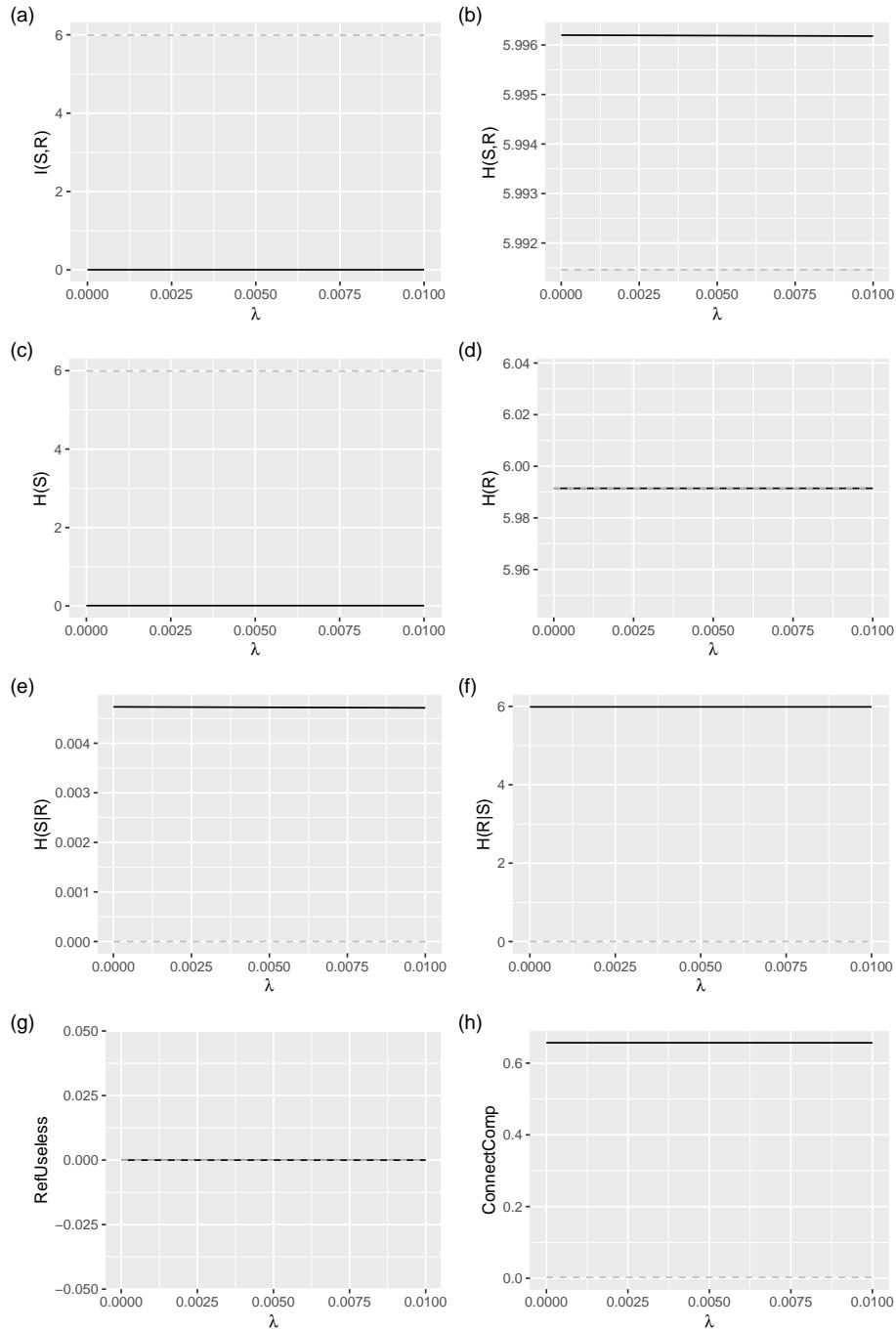


Figure 4.22: this happens for $\phi = 1$ but not for $\phi = 0$

second model, $\phi = 1$

all π uniform

- allow disconnect meanings

- initial: random 4.17
- initial: single 4.18
- initial: one-to-one 4.19

- do not allow disconnect meanings 

- initial: random 4.20
- initial: single 4.21
- initial: one-to-one 4.22

4.2.2 Inside selected lambda

first model, $\phi = 0$

- initial: random 4.23
- initial: single 4.24
- initial: one-to-one 4.25

first model, $\phi = 1$

- initial: random 4.26
- initial: single 4.27
- initial: one-to-one 4.28

second model, $\phi = 0$

all π uniform

- allow disconnect meanings

- initial: random 4.29
- initial: single 4.30
- initial: one-to-one 4.31

- do not allow disconnect meanings 

- initial: random 4.32
- initial: single 4.33
- initial: one-to-one 4.34

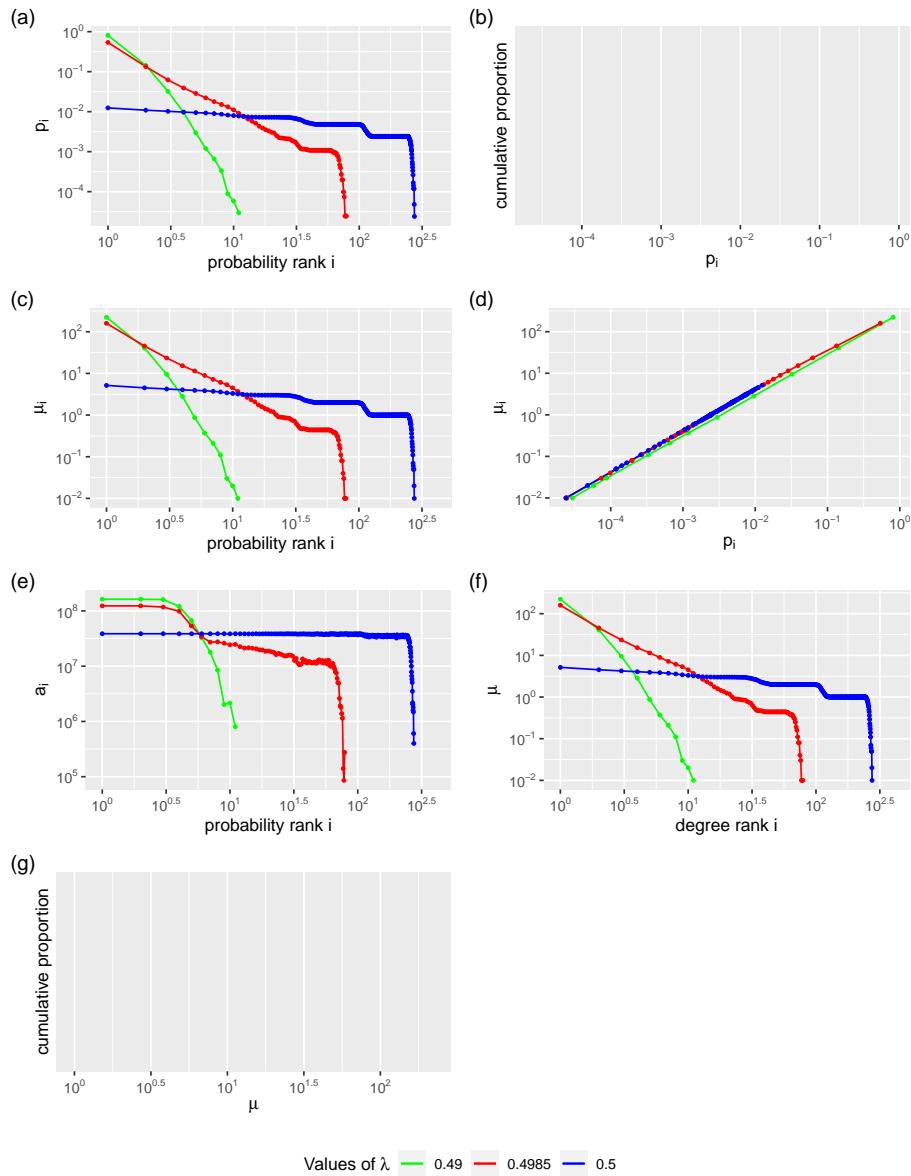


Figure 4.23: a

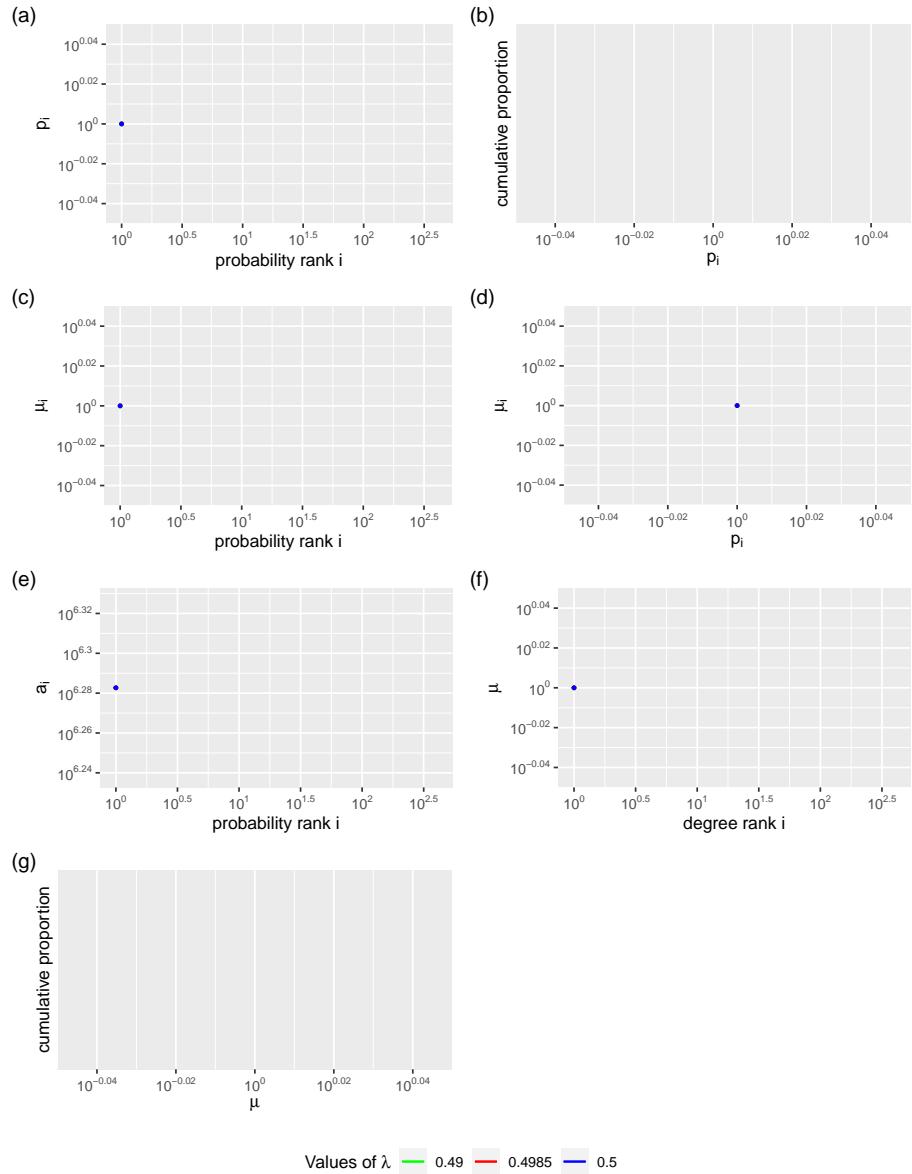


Figure 4.24: a

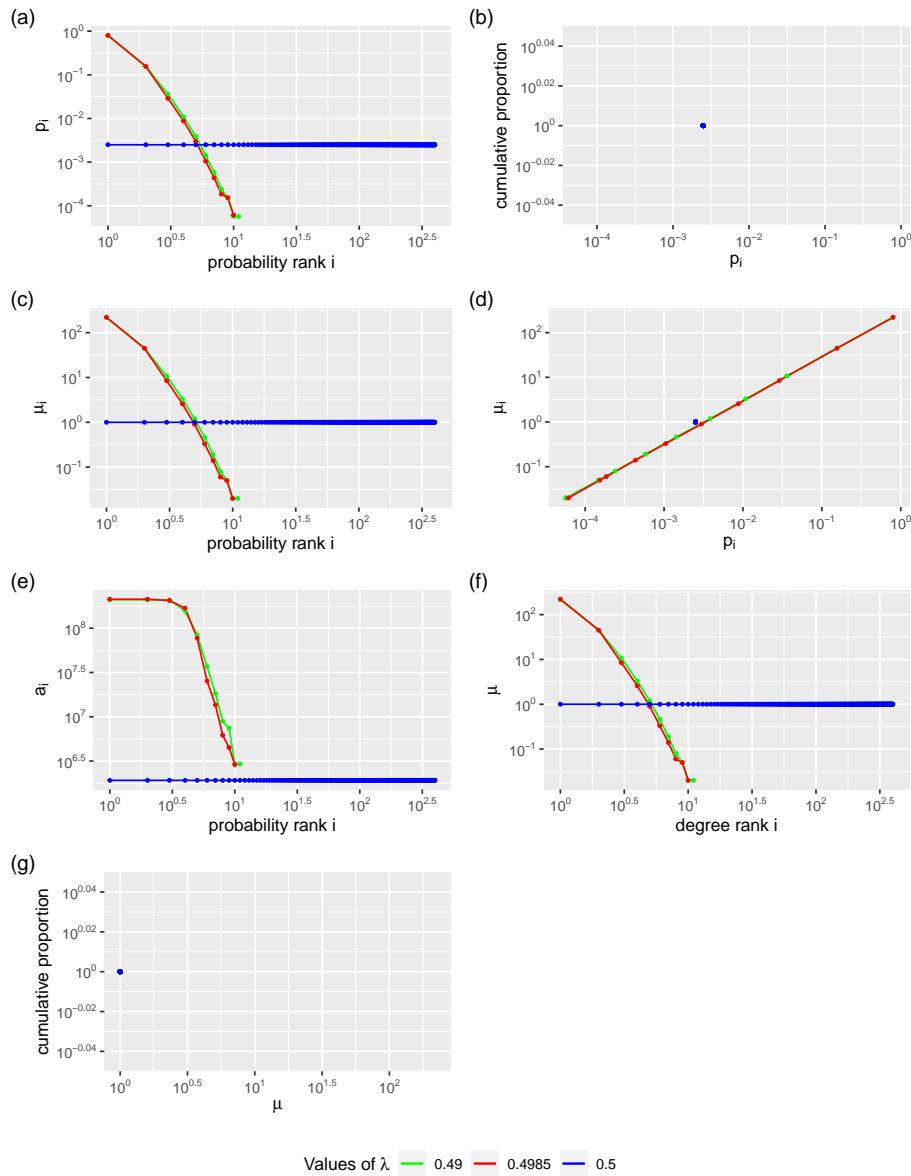


Figure 4.25: a

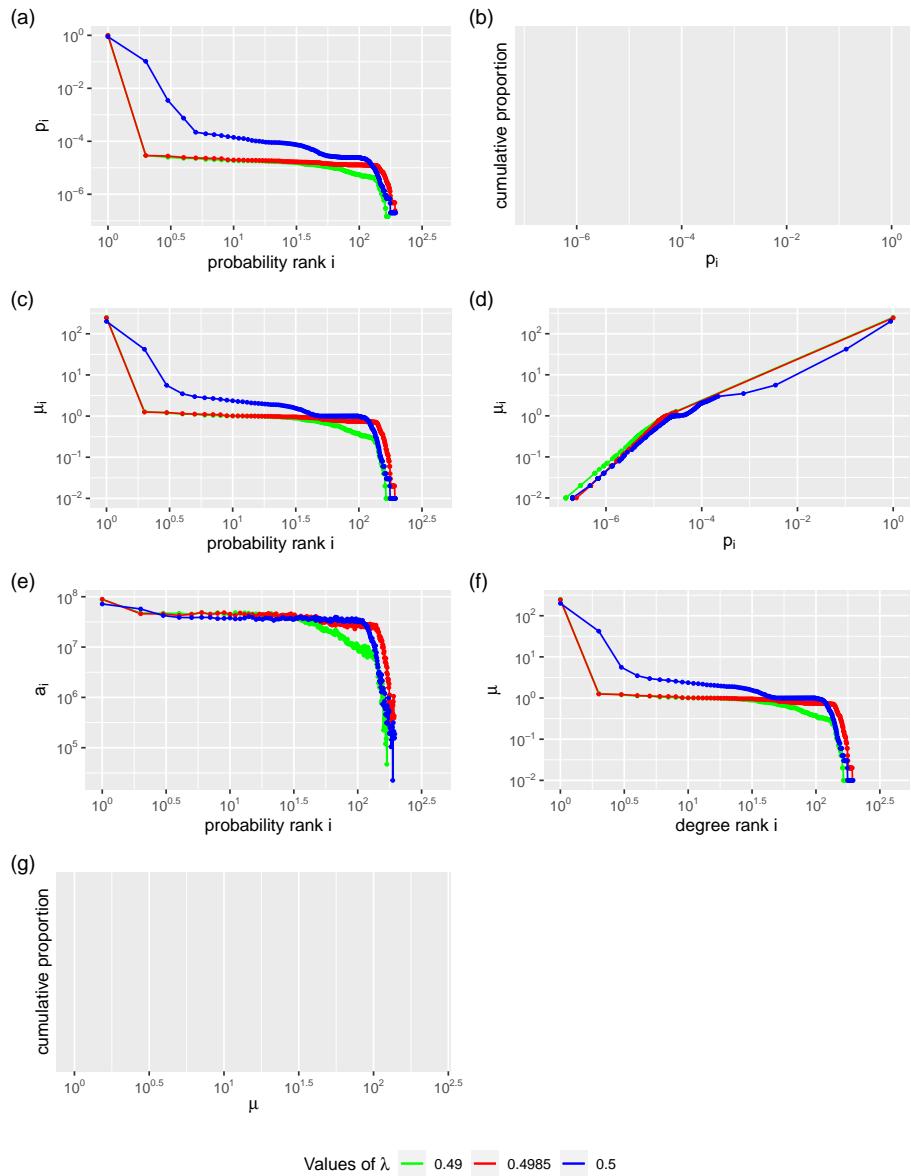


Figure 4.26: a

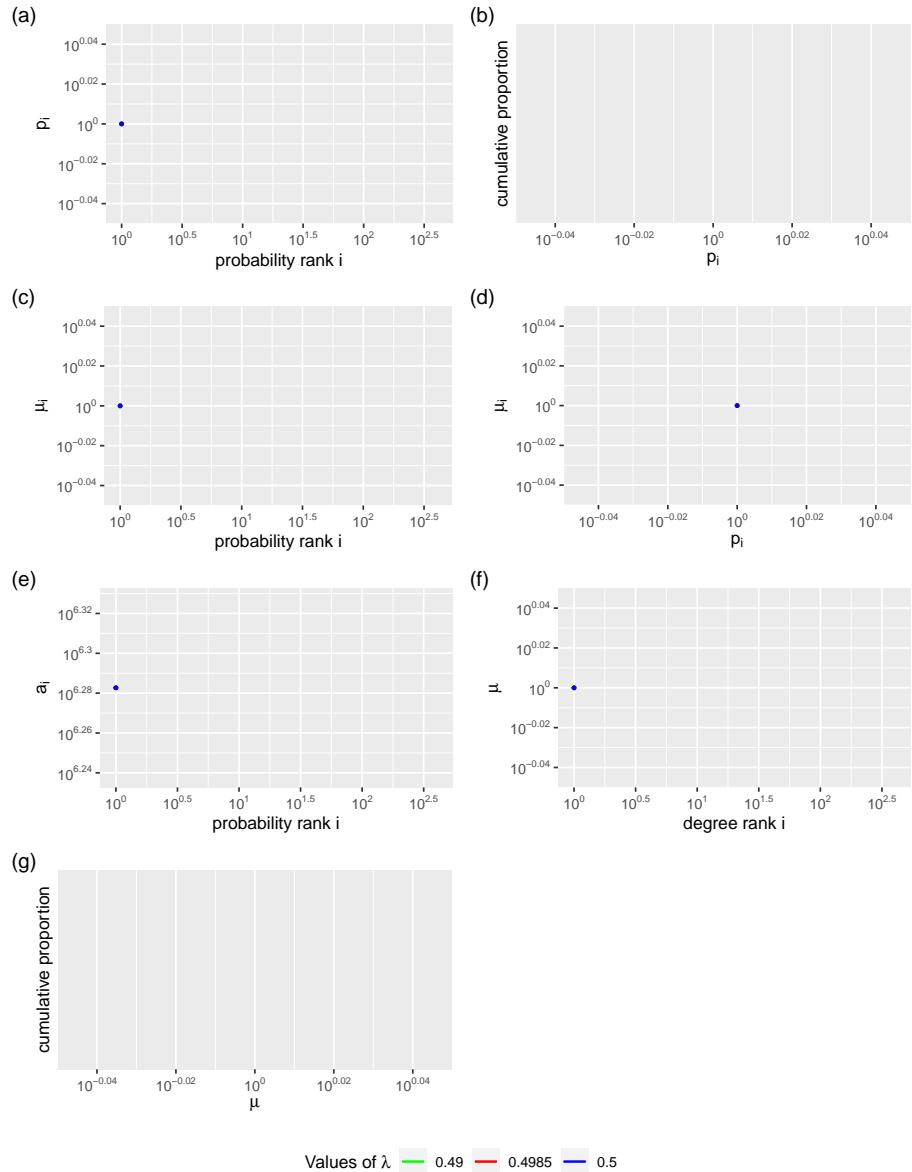


Figure 4.27: a

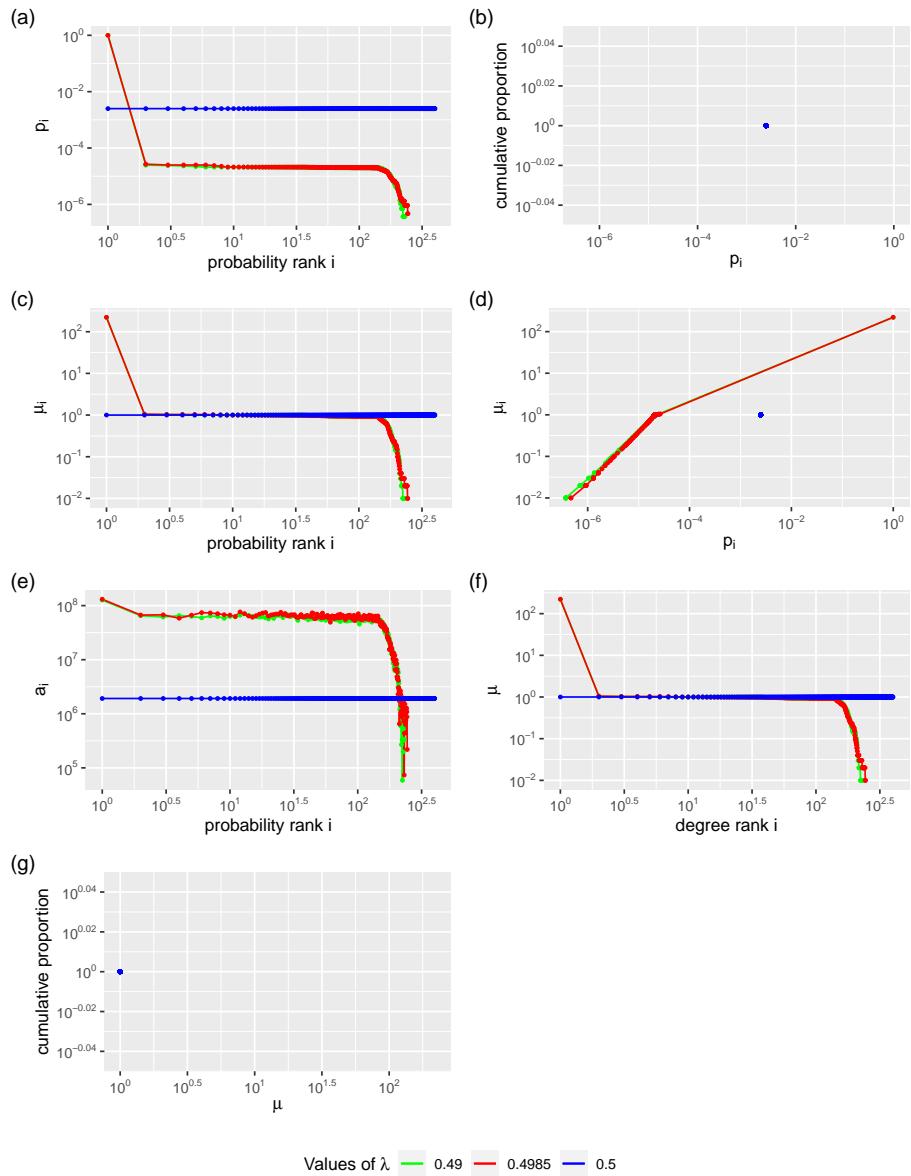


Figure 4.28: a

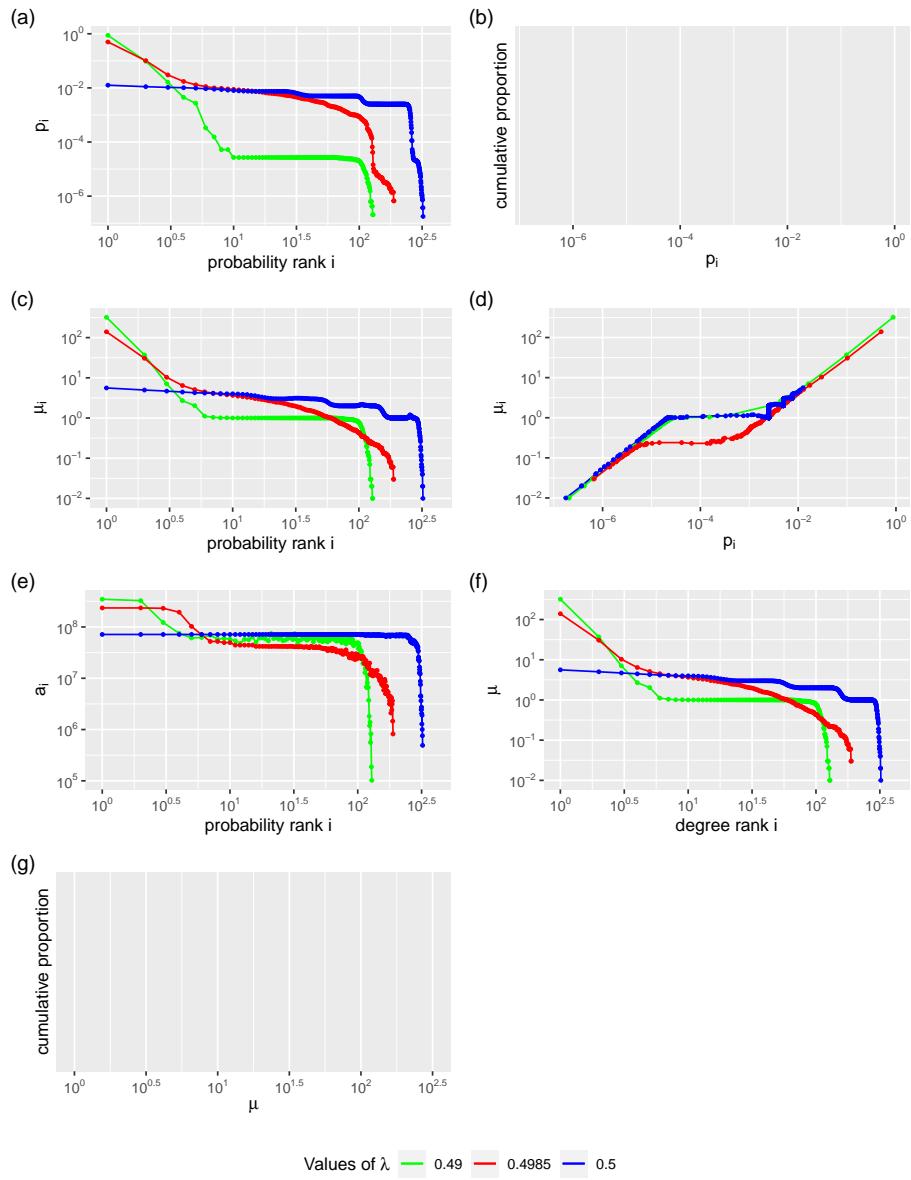


Figure 4.29: a

insideLambda_uniform_phi0_nm400_dynamic_singleLink_allowUnlinked.pdf

Figure 4.30: a

insideLambda_uniform_phi0_nm400_dynamic_oneToOne_allowUnlinked.pdf

Figure 4.31: a

insideLambda_uniform_phi0_nm400_dynamic_randomBipartite_disallowUnlinked.pdf

Figure 4.32: a

insideLambda_uniform_phi0_nm400_dynamic_singleLink_disallowUnlinked.pdf

Figure 4.33: a

insideLambda_uniform_phi0_nm400_dynamic_oneToOne_disallowUnlinked.pdf

Figure 4.34: a

insideLambda_uniform_phi1_nm400_dynamic_randomBipartite_allowUnlinked.pdf

Figure 4.35: a

insideLambda_uniform_phi1_nm400_dynamic_singleLink_allowUnlinked.pdf

Figure 4.36: a

insideLambda_uniform_phi1_nm400_dynamic_oneToOne_allowUnlinked.pdf

Figure 4.37: a

insideLambda_uniform_phi1_nm400_dynamic_randomBipartite_disallowUnlinked.pdf

Figure 4.38: a

insideLambda_uniform_phi1_nm400_dynamic_singleLink_disallowUnlinked.pdf

Figure 4.39: a

insideLambda_uniform_phi1_nm400_dynamic_oneToOne_disallowUnlinked.pdf

Figure 4.40: a

second model, $\phi = 1$

all π uniform

- allow disconnect meanings

- initial: random 4.35
- initial: single 4.36
- initial: one-to-one 4.37

- do not allow disconnect meanings 

- initial: random 4.38
- initial: single 4.39
- initial: one-to-one 4.40

4.2.3 Graph visualization (smaller graphs)

all $\phi = 1$



first model

- initial: random 4.41
- initial: single 4.42
- initial: one-to-one 4.43
- initial: complete 4.44

second model

- initial: random 4.45
- initial: single 4.46
- initial: one-to-one 4.47
- initial: complete 4.48

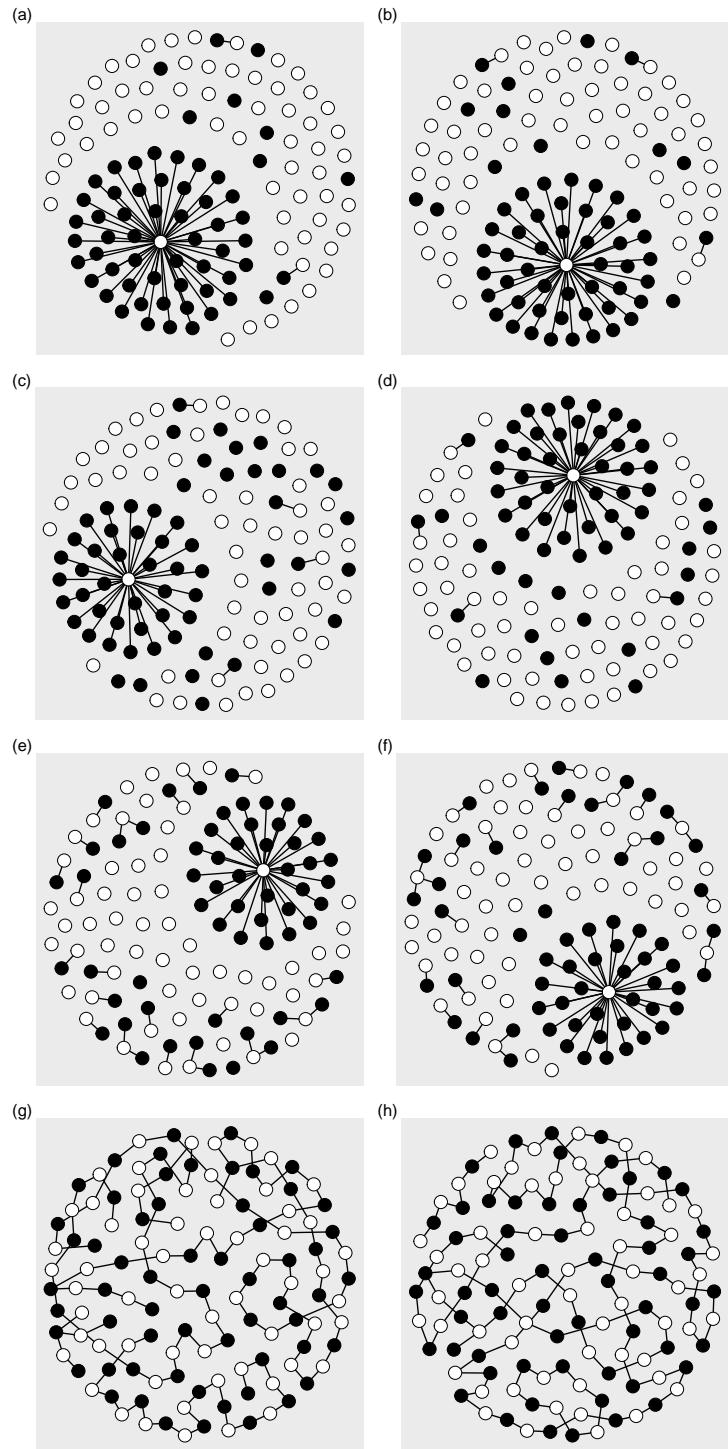


Figure 4.41: a
82

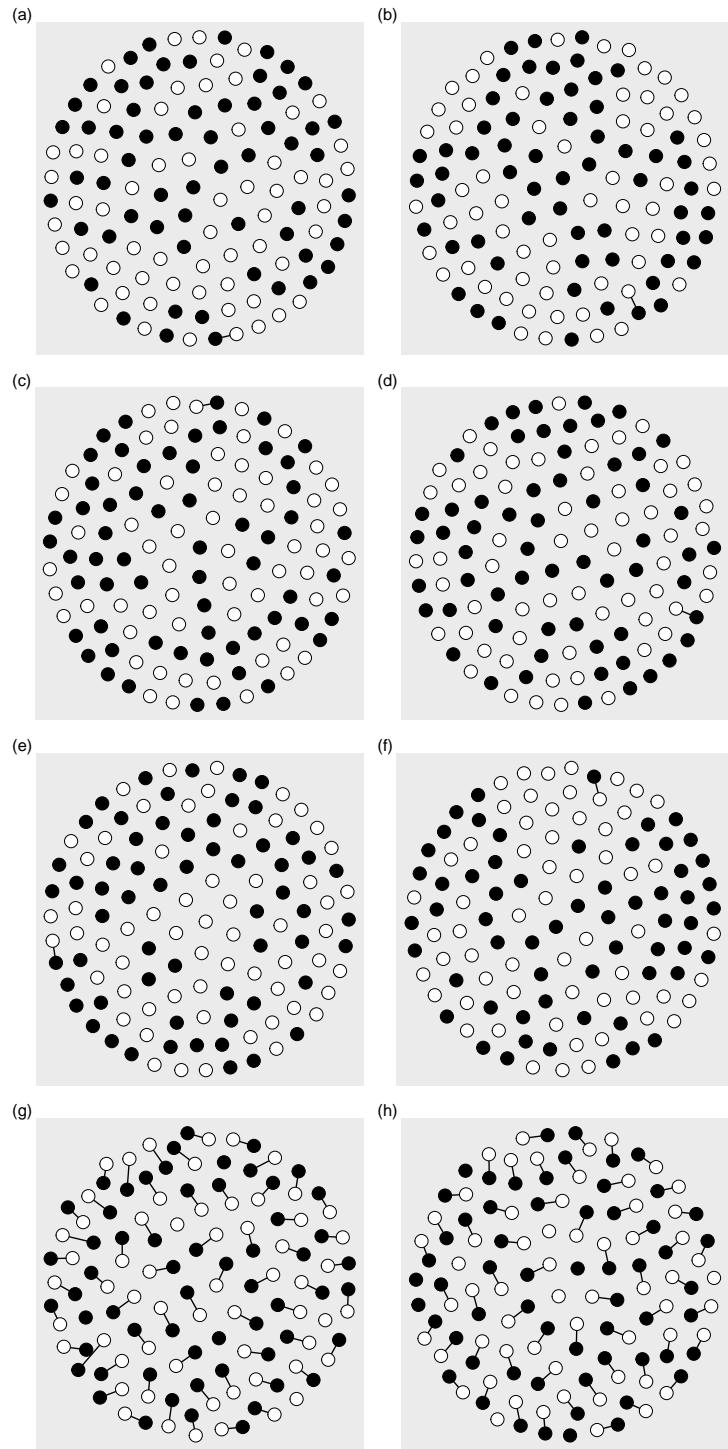


Figure 4.42: a
83

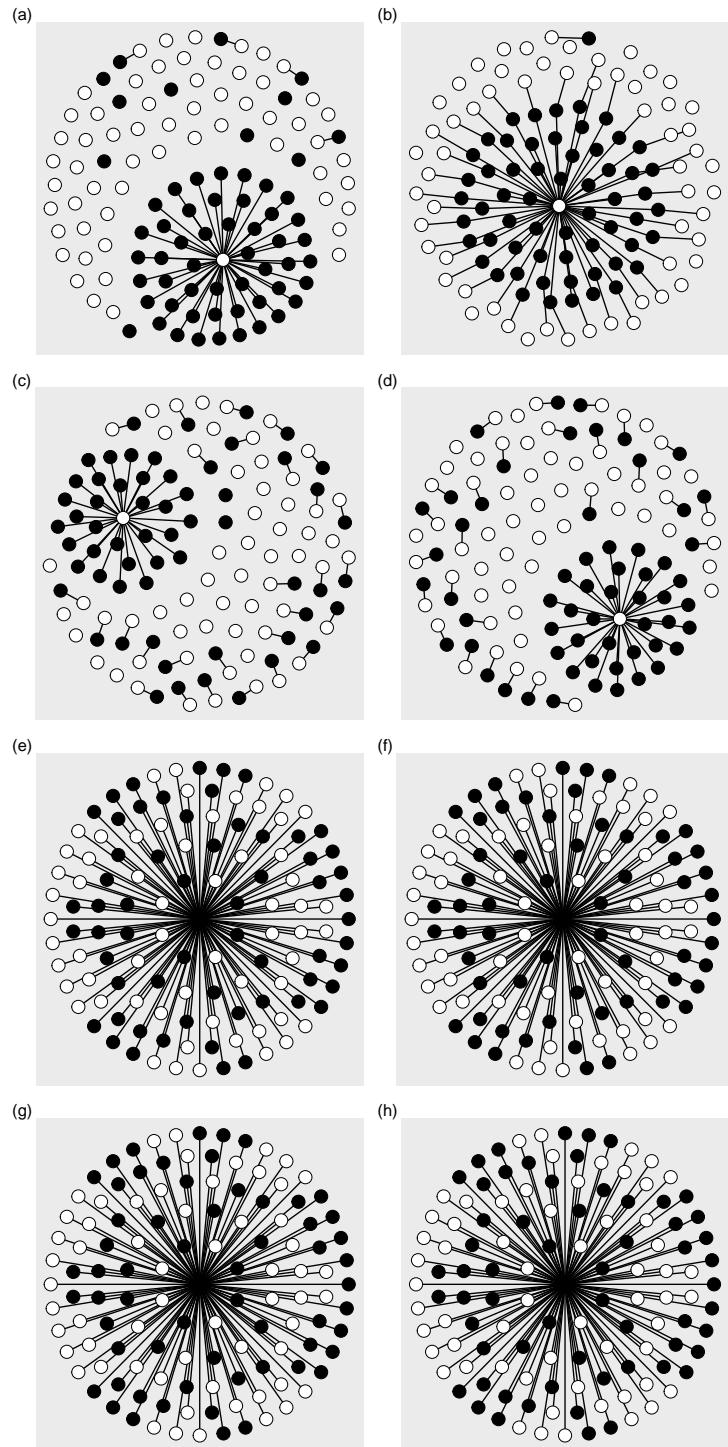


Figure 4.43: a
84

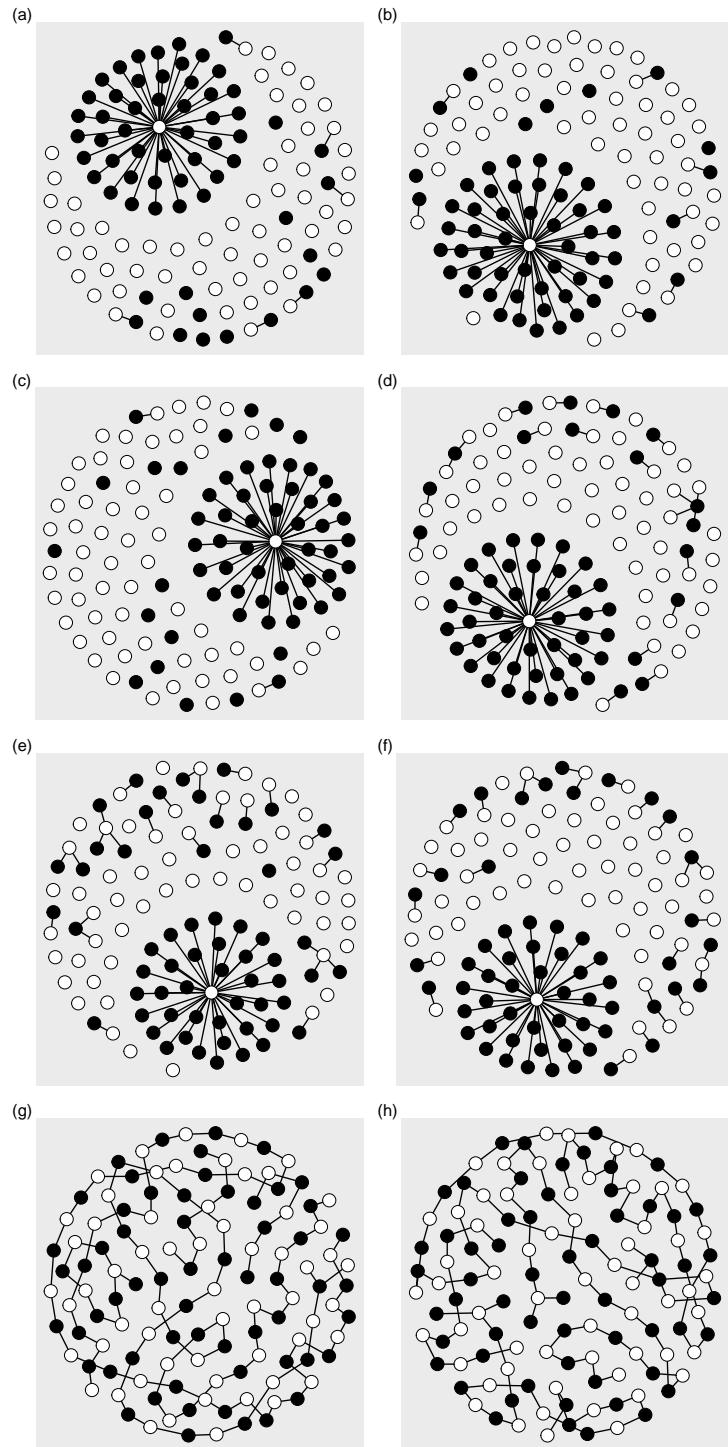


Figure 4.44: a
85

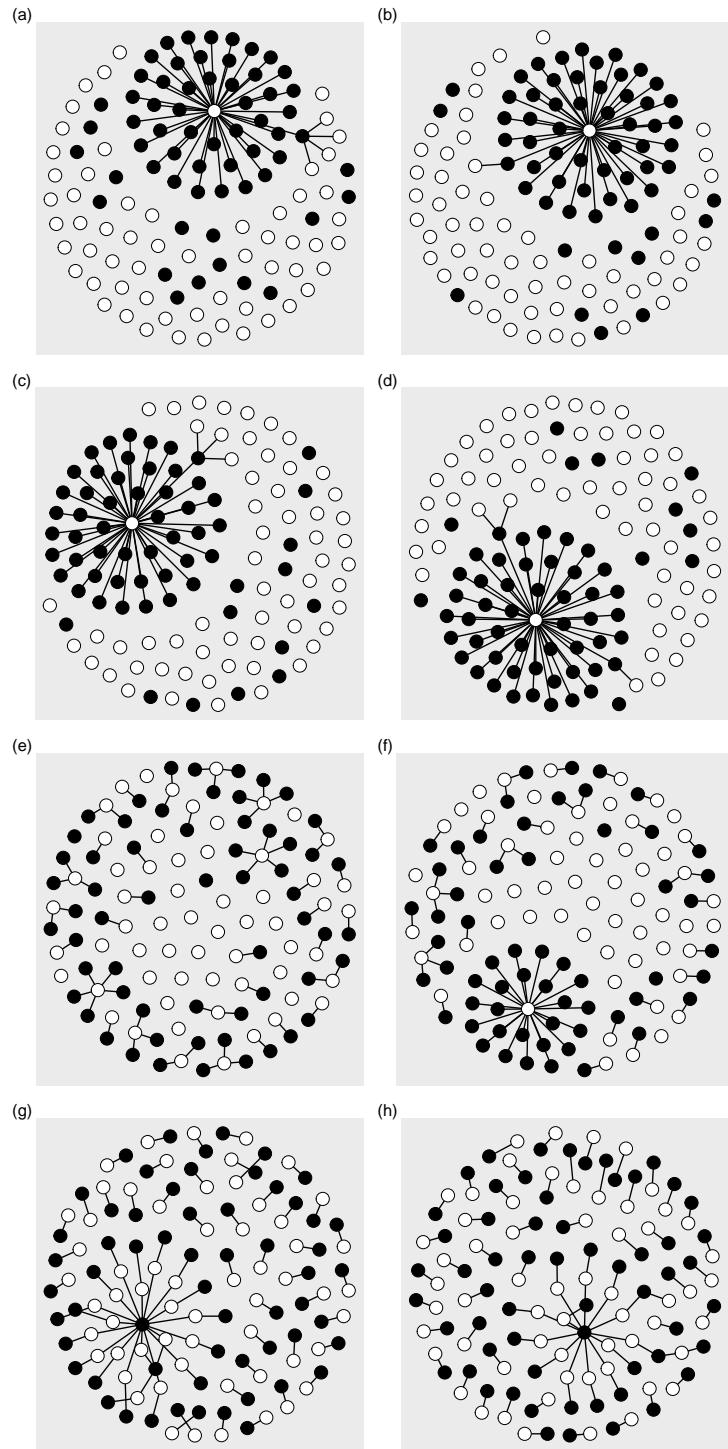


Figure 4.45: a
86

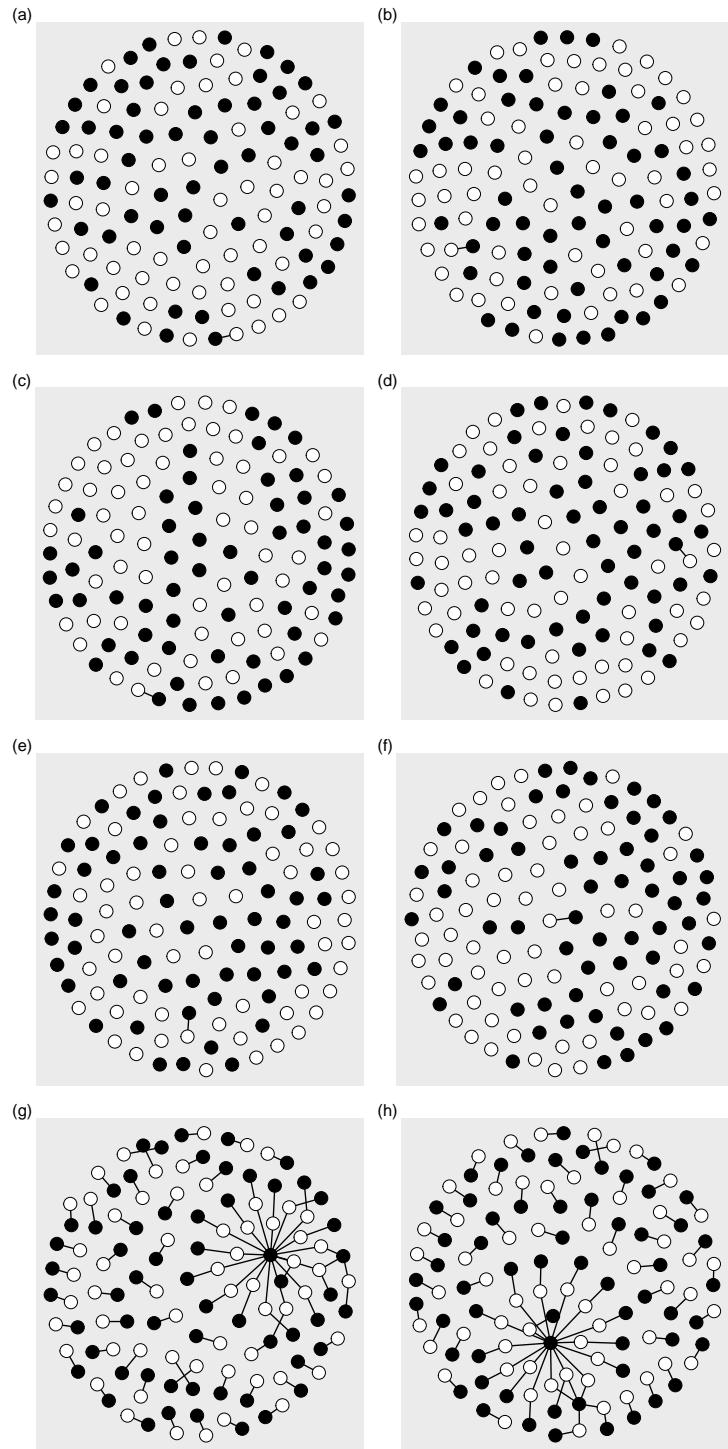


Figure 4.46: a
87

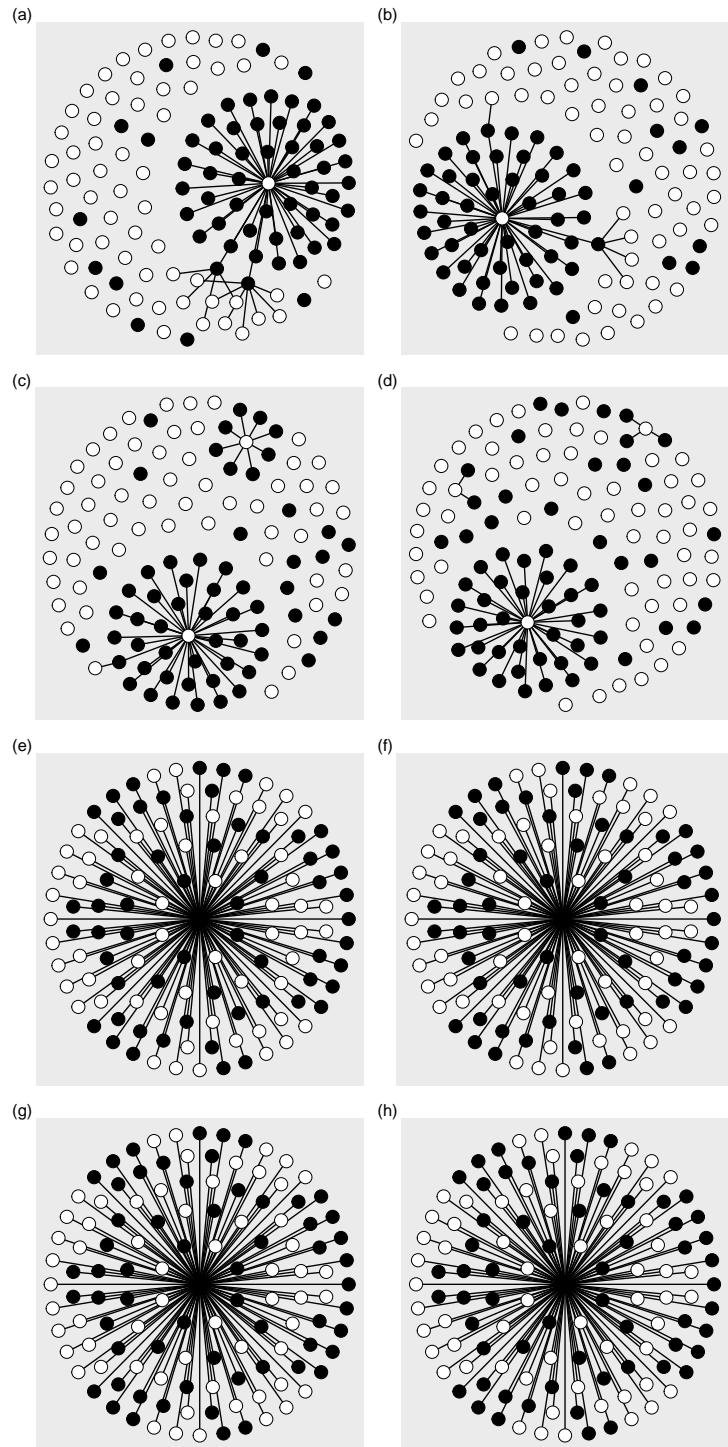


Figure 4.47: a
88

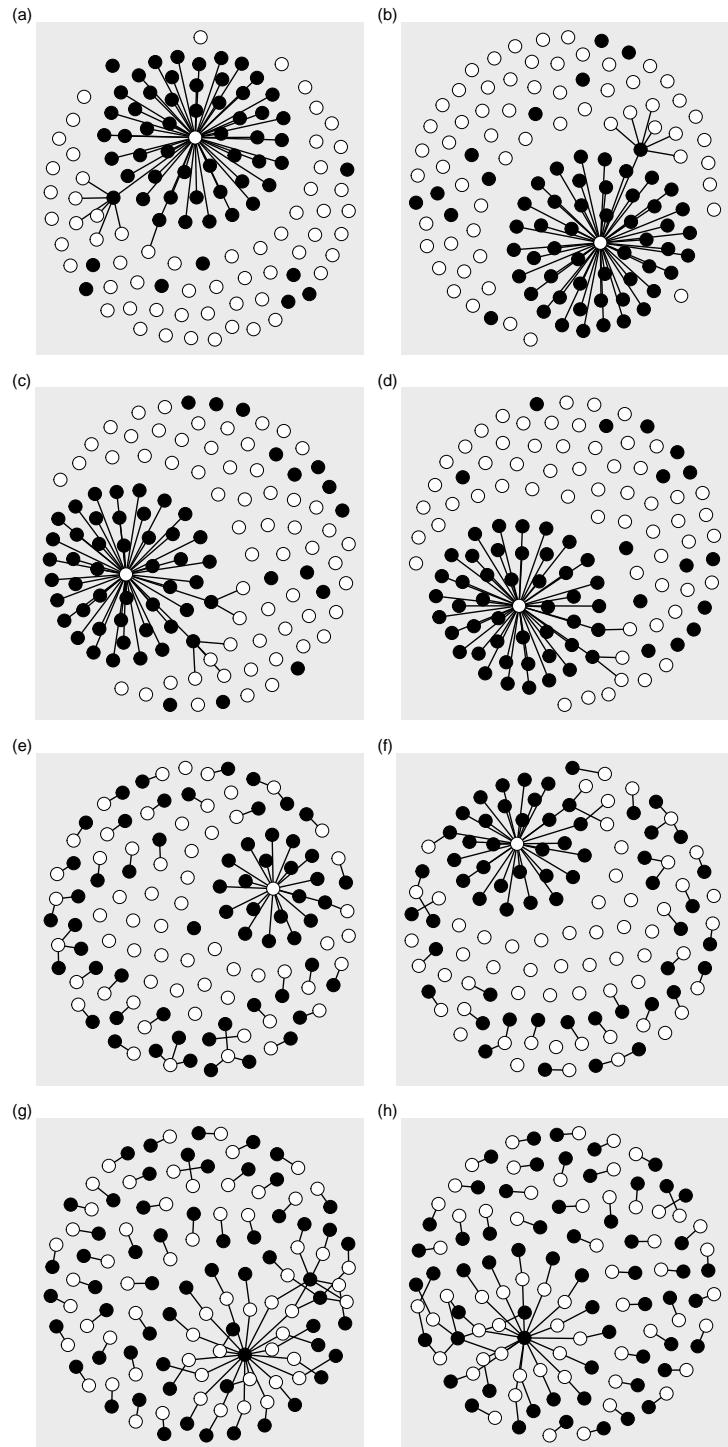


Figure 4.48: a
89

Chapter 5

Discussion

Aquesta part és important, ha de quedar clara. Parlar també sobre limitacions: quins resultats son contundents i quins febles.

Analisi critic, interpretar, lligar amb context de coneixements: relació paraula edat freqüència i altres models de zipf com es relacionen amb ells, com ho expliquen. Distinció entre model que explica i model que descriu. No hi ha explicació sense teoria. Teoria és una sèrie de principis. No tots els models son explicacions però no poden fer prediccions. E.g., random typing no diu res de significats no pot explicar aprenentatge en nens.

5.1 mètodes (lingüística quantitativa)

5.1.1 zipf law en regressió lineal del log-log?

- mirar gràfiques dels dos models, variants

5.1.2 relació edat-freqüència

- gràfiques mostren que paraules més antigues són més freqüents
- corresponen amb zipf (llibre)



5.2 computacional

- lligar amb biosemiotics

5.2.1 mínim local

- el llei zipf mínim local? stop condition
- fins a quin punt recuperar lleis depenen de paràmetres?

5.3 future work

5.3.1 altres metodes d'optimitzacio

- simulated annealing (non zero temperature)
- gradient descent (make Ω derivable)

5.3.2 altres valors de phi

hem fet 1 i 0 per altres, 0.5, 1.5...

5.3.3 vocabulary nens segon model

biosemiotics ha fet primer model nomes

5.3.4 evitar error numeric encara mes

- subcalculs del metode dinamic que es podrien fer estatics
- ja es fa un valor de χ

Appendix A

Code

A.1 repo

- afegir repositori amb el codi

A.2 Program parameters

Here all the parameters that can be specified to the tool created as part of this thesis are explained.

- n and m , the size of the two sets in the bipartite graph
- ϕ the parameter that was added to generalize the older models
- Whether or not to use constant *a priori* probabilities for the meanings. Using them implies using the external model, not using them means using the internal model.
- Whether unlinked objects are allowed or not.
- The π parameter, the distribution of the *a priori* probabilities, can be specified in a number of ways if these probabilities are indeed being used. In addition to those specified in Section 3.1, it is allowed to manually specify an arbitrary probability distribution.
- Which λ parameters to test, the program will generate data for all the specified λ parameters. They can be specified either as a range or manually one by one.
- The number of realizations to be done. For each value of λ , that many optimization processes will be carried out and the results averaged.

- The initial graph that the optimization starts from. This can be a $G_{n,m,p}$ or $G_{n,m,e}$ random bipartite graph (see Section 3.1), a complete graph or a one to one bijection. In that last case, only the first $\min(n, m)$ words and meanings are connected.
- Whether to use the static or dynamic equations.
- The number of mutations. Either a constant number of or random binomial mutations with a given probability.
- The stop condition. Either of the weak or strong conditions, as well as the option of manually specifying a number. It is always the number of failures to improve Ω .
- A random seed can be manually specified or left random (even if random, the seed used is always logged). Whether to emit or not certain results. By default the program emits several csv files corresponding to the information theoretical plots as a function of lambda and the statistics for every value of lambda computed. It also outputs every generated graph in every realization. Any other measure could be obtained from these graphs.

Appendix B

Formulae

derivacions llargues i pagines i pagines de mates van aqui

B.1 Properties

B.1.1 Simplification of the equations of entropies

Property 1 is used in several parts of the thesis to simplify the expressions of entropies. The derivation is short, but it is not added into the main text of the thesis for the sake of clarity and focus.

$$\begin{aligned} & - \sum_i \frac{x_i}{T} \log \frac{x_i}{T} \\ & - \frac{1}{T} \sum_i x_i \log x_i - x_i \log T \\ & - \frac{1}{T} \sum_i x_i \log x_i + \log(T) \frac{1}{T} \sum_i x_i \end{aligned}$$

at this point we can apply

$$\sum_i x_i = T$$

and obtain

$$\log T - \frac{1}{T} \sum_i x_i \log x_i$$

B.1.2 Proof that $\sum_{i=1}^n \mu_i^\phi \chi_i = \rho$

In order to use Property 1 to simplify the derivation of Equation (2.44), Property 2 must hold. Here it is shown that this is the case.

Starting from the equality, we can expand both sides

$$\begin{aligned} \sum_{i=1}^n \mu_i^\phi \chi_i &= \rho \\ \sum_{i=1}^n \mu_i^\phi \sum_{j=1}^m \frac{a_{ij}(1 - \delta_{\omega_j,0})\pi(r_j)}{\omega_{\phi,j}} &= \sum_{j=1}^m (1 - \delta_{\omega_j,0})\pi(r_j) \\ \sum_{j=1}^m \left(\sum_{i=1}^n a_{ij} \mu_i^\phi \right) \frac{(1 - \delta_{\omega_j,0})\pi(r_j)}{\omega_{\phi,j}} &= \sum_{j=1}^m (1 - \delta_{\omega_j,0})\pi(r_j), \end{aligned}$$

applying Equation (2.4) we obtain

$$\sum_{j=1}^m (1 - \delta_{\omega_j,0})\pi(r_j)$$

on both sides.

Recall the definition of $p(s_i)$ in Equation (2.39) to see that

$$\sum_{i=1}^n p(s_i) = 1$$

immediately follows.

B.2 Derivation of the joint entropy in the external model

In Section 2.1.3 the derivation of the joint entropy is left as just the high level reasoning of which formulas may be used to obtain its expression. Here, the full derivation of both approaches is given.

The first approach consists of applying Equation (2.38) to the information

theory definition of $H(S, R)$ (Equation (1.7)),

$$\begin{aligned}
H(S, R) &= - \sum_{i=1}^n \sum_{j=1}^m p(s_i, r_j) \log [p(s_i, r_j)] \\
&= - \sum_{i=1}^n \sum_{j=1}^m \frac{a_{i,j}(1 - \delta_{\omega_j, 0})\mu_i^\phi \pi(r_j)}{\rho \omega_{\phi, j}} \log \left[\frac{a_{i,j}(1 - \delta_{\omega_j, 0})\mu_i^\phi \pi(r_j)}{\rho \omega_{\phi, j}} \right] \\
&= - \frac{1}{\rho} \sum_{i=1}^n \sum_{j=1}^m \frac{a_{i,j}(1 - \delta_{\omega_j, 0})\mu_i^\phi \pi(r_j)}{\omega_{\phi, j}} \log \left[\frac{\mu_i^\phi \pi(r_j)}{\rho \omega_{\phi, j}} \right] \\
&= - \frac{1}{\rho} \sum_{j=1}^m \frac{(1 - \delta_{\omega_j, 0})\pi(r_j)}{\omega_{\phi, j}} \left[\phi \sum_{i=1}^n a_{i,j} \mu_i^\phi \log \mu_i \right. \\
&\quad + \log \pi(r_j) \sum_{i=1}^n a_{i,j} \mu_i^\phi - \log(\rho) \sum_{i=1}^n a_{i,j} \mu_i^\phi \\
&\quad \left. - \log(\omega_{\phi, j}) \sum_{i=1}^n a_{i,j} \mu_i^\phi \right].
\end{aligned}$$

Applying Equation (2.43) we arrive at Equation (2.42).

The second approach consists on using other less complex expressions we have available and building up from them using the definitions

$$H(S, R) = H(S|R) + H(R), \quad (\text{B.1})$$

$$H(S|R) = \sum_{j=1}^m H(S|r_j)p(r_j), \quad (\text{B.2})$$

$$H(S|r_j) = - \sum_{i=1}^n p(s_i|r_j) \log p(s_i|r_j). \quad (\text{B.3})$$

Starting from Equation (B.3) and applying Equation (2.37) we obtain

$$\begin{aligned}
H(S|r_j) &= - \sum_{i=1}^n \frac{a_{i,j}\mu_i^\phi \pi(r_j)}{\rho \omega_{\phi, j}} \log \left(\frac{a_{i,j}\mu_i^\phi \pi(r_j)}{\rho \omega_{\phi, j}} \right) \\
&= \log \omega_{\phi, j} - \frac{1}{\omega_{\phi, j}} \sum_{i=1}^n a_{i,j} \mu_i^\phi \log(a_{i,j} \mu_i^\phi) \\
&= \log \omega_{\phi, j} - \frac{\phi}{\omega_{\phi, j}} \sum_{i=1}^n a_{i,j} \mu_i^\phi \log(\mu_i)
\end{aligned}$$

and after applying Equation (2.43)

$$H(S|r_j) = \log \omega_{\phi, j} - \frac{\phi \nu_j}{\omega_{\phi, j}}. \quad (\text{B.4})$$

Applying Equations (B.4) and (2.35) to Equation (B.2) we reach

$$\begin{aligned} H(S|R) &= \sum_{j=1}^m \left(\log \omega_{\phi,j} - \frac{\phi \nu_j}{\omega_{\phi,j}} \right) \frac{(1 - \delta_{w_j,0}) \pi(r_j)}{\rho} \\ &= \frac{1}{\rho} \sum_{j=1}^m (1 - \delta_{w_j,0}) \pi(r_j) \left[\log(\omega_{\phi,j}) - \frac{\phi \nu_j}{\omega_{\phi,j}} \right]. \end{aligned} \quad (\text{B.5})$$

The last step consists of applying Equations (B.5) and (2.41) to Equation (B.1), obtaining Equation (2.42) again

$$\begin{aligned} H(S, R) &= \frac{1}{\rho} \sum_{j=1}^m (1 - \delta_{w_j,0}) \pi(r_j) \left[\log(\omega_{\phi,j}) - \frac{\phi \nu_j}{\omega_{\phi,j}} \right] \\ &\quad + \log \rho - \frac{1}{\rho} \sum_{j=1}^m (1 - \delta_{w_j,0}) \pi(r_j) \log \pi(r_j) \\ &= \log \rho - \frac{1}{\rho} \sum_{j=1}^m (1 - \delta_{w_j,0}) \pi(r_j) \left[\frac{\phi \nu_j}{\omega_{\phi,j}} + \log \frac{\pi(r_j)}{\omega_{\phi,j}} \right]. \end{aligned}$$

B.3 Dynamic Equations for the external model

The full derivation of the dynamic equations for the external model is given in this section. The equations themselves as well as other definitions used here are already given in Section 2.1.3.

In this section, the derivation is logical rather than mathematical, consisting of explanations behind the logic of the equations.

ρ (see Equation (2.36)) only changes when, as a result of the mutation, the meaning r_j either was connected and becomes disconnected or was disconnected and becomes connected, resulting in Equation (2.55).

ν_l is very similar to $\omega_{\phi,l}$ (compare Equations (2.43) and (2.4)). Equation (2.56) shows how ν_l changes in the same cases and in the same way as $\omega_{\phi,l}$ (Equation (2.29)).

χ_k depends on $\omega_{\phi,l}$ such that $l \in \Gamma_S(k)$ (see Equation (2.40)). For χ_i , the entire value has to be recalculated, as every $\omega_{\phi,l}$ for $l \in \Gamma_S(k)$ will have changed (Equation (2.29)). It is more efficient (and reduces the amount of floating point error) to calculate χ_i statically than to subtract every $\omega_{\phi,l}$ and then add every $\omega'_{\phi,l}$. For all other values of χ_k ($k \neq i$), the affected $\omega_{\phi,l}$ are the intersection of the set of neighbors of i plus the meaning r_j and the set of neighbors of k . This is the set $A_{i,j}(k)$ defined in Equation (2.53). With this reasoning we reach the formula for the dynamic recalculation of χ_k , Equation (2.57).

$X(R)$ (Equation (2.50)) changes in a very similar way to ρ (Equation (2.36)), as they both depend on the same variable in the same way: only when the meaning r_j becomes connected or disconnected does $X(R)$ change, resulting in Equation (2.55).

$X(S, R)$ (Equation (2.48)) will change when a $x(r_l)$ (Equation (2.52)) changes. Both $\omega_{\phi,l}$ and ν_l change for $l \in \Gamma_S(i) \cup \{j\}$ (Equations (2.29) and (2.56)). So these are the components $x(s_l)$ of $H(S, R)$ that will change. For the special case of $l = j$ we should not subtract the old value when r_j has become connected as a result of the mutation (as that component was not present), and we should not add the new value when r_j has become disconnected (as that component should not be present). Or in other words, only subtract the old value if r_j was connected before the mutation (this serves to either update it or to remove it) and only add the new value if r_j is connected after the mutation (this serves to either update it or to add it). The resulting expression for $X'(S, R)$ is in Equation (2.59).

$X(S)$ (Equation (2.49)) will change when $x(s_k)$ (Equation (2.51)) changes, which depends on χ_k (Equation (2.57)). χ_k is updated for every k such that $|A_k| \neq \emptyset$ and χ_i is always updated, which is the definition of the $B_{i,j}(k)$ set (Equation (2.54)). Using $B_{i,j}(k)$ we obtain the expression for $X'(S)$ in Equation (2.60).

Bibliography

- [1] Andrea Baronchelli et al. “Sharp transition towards shared vocabularies in multi-agent systems”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2006.06 (2006), P06014.
- [2] Luciana S Buriol, Mauricio GC Resende, and Mikkel Thorup. “Speeding up dynamic shortest path algorithms”. In: *Rapport technique TD-5RJ8B, AT&T Labs Research* (2003), p. 41.
- [3] David Carrera-Casado and Ramon Ferrer-i-Cancho. “The advent and fall of a vocabulary learning bias from communicative efficiency”. In: *Biosemiotics* (2021). URL: <http://arxiv.org/abs/2105.11519>.
- [4] Noam Chomsky and George Miller. “Finitary models of language users”. In: *Handbook of mathematical psychology* 2 (1963), pp. 419–491.
- [5] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.
- [6] Paul Erdős and Alfréd Rényi. “On the evolution of random graphs”. In: *Publ. Math* 5.1 (1960), pp. 17–60.
- [7] Ramon Ferrer-i-Cancho. “Compression and the origins of Zipf’s law for word frequencies”. In: *Complexity* 21 (2016), pp. 409–411. DOI: 10.1002/cplx.21820.
- [8] Ramon Ferrer-i-Cancho. “The optimality of attaching unlinked labels to unlinked meanings”. In: *Glottometrics* 36 (2017), pp. 1–16.
- [9] Ramon Ferrer-i-Cancho. “Zipf’s law from a communicative phase transition”. In: *European Physical Journal B* 47.3 (2005), pp. 449–457. DOI: 10.1140/epjb/e2005-00340-y.
- [10] Ramon Ferrer-i-Cancho, Christian Bentz, and Caio Seguin. “Optimal Coding and the Origins of Zipfian Laws”. In: *Journal of Quantitative Linguistics* (2020). DOI: 10.1080/09296174.2020.1778387.
- [11] Ramon Ferrer-i-Cancho and Albert Díaz-Guilera. “The global minima of the communicative energy of natural communication systems”. In: *Journal of Statistical Mechanics: Theory and Experiment* 06009.6 (2007). DOI: 10.1088/1742-5468/2007/06/P06009.

- [12] Ramon Ferrer-i-Cancho and Ricard V. Sole. “Least effort and the origins of scaling in human language”. In: *Proceedings of the National Academy of Sciences of the United States of America* 100.3 (2003), pp. 788–791. DOI: 10.1073/pnas.0335980100.
- [13] Ramon Ferrer-i-Cancho and Michael S. Vitevitch. “The origins of Zipf’s meaning-frequency law”. In: *Journal of the Association for Information Science and Technology* 69.11 (Nov. 2018), pp. 1369–1379. DOI: 10.1002/asi.24057.
- [14] Ali Mehri and Maryam Jamaati. “Variation of Zipf’s exponent in one hundred live languages: A study of the Holy Bible translations”. In: *Physics Letters A* 381.31 (2017), pp. 2470–2477. DOI: 10.1016/j.physleta.2017.05.061.
- [15] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2017.
- [16] Mikhail Prokopenko et al. “Phase transitions in least-effort communications”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2010.11 (2010). DOI: 10.1088/1742-5468/2010/11/P11025.
- [17] Terry Regier, Paul Kay, and Naveen Khetarpal. “Color naming reflects optimal partitions of color space”. In: *Proceedings of the National Academy of Sciences of the United States of America* 104.4 (2007), pp. 1436–1441. DOI: 10.1073/pnas.0610341104.
- [18] Christoph Salge et al. “Zipf’s law: Balancing signal usage cost and communication efficiency”. In: *PLoS ONE* 10.10 (2015), pp. 1–14. DOI: 10.1371/journal.pone.0139475.
- [19] Sergio Jimenez. *Zipf’s law*. [Online; accessed September 26, 2021]. 2015. URL: https://commons.wikimedia.org/wiki/File:Zipf_30wiki_en_labels.png.
- [20] Herbert A Simon. “On a Class of Skew Distribution Functions”. In: *Biometrika* 42.3/4 (1955), p. 425. DOI: 10.2307/2333389.
- [21] J. S. Smart. “Statistical tests of the broken-stick model of species-abundance relations”. In: *Journal of Theoretical Biology* 59.1 (1976), pp. 127–139. DOI: 10.1016/S0022-5193(76)80027-6.
- [22] G. K. Zipf. *Human behaviour and the principle of least effort*. Cambridge (MA), USA: Addison-Wesley, 1949.