Entropy-based methods to tackle missing information in complex networks.

PhD Program in Economics, Management and Data Science

XXXI Cycle

By

Federica Parisi

2018

The dissertation of Federica Parisi is approved.

Program Coordinator: Prof. Dr. Massimo Riccaboni, IMT School for Advanced Studies Lucca

Supervisor: Prof. Dr. Guido Caldarelli, IMT School for Advanced Studies Lucca

Supervisor: Dr. Tiziano Squartini, IMT School for Advanced Studies Lucca

The dissertation of Federica Parisi has been reviewed by:

Dr. Andrea Gabrielli, Institute of Complex Systems - CNR, Rome "Sapienza" University

Prof. Dr. Claudio J. Tessone, University of Zurich

Contents

List of Figures **x**

List of Tables **xii**

Acknowledgements **xiii**

Vita and Publications **xiv**

Abstract **xvi**

1 Introduction **1**

1.1 Basic definitions and notation 5

1.1.1 Bipartite graphs 8

1.2 Graph Ensembles and the ERG model 9

1.2.1 The Exponential Random Graph model 11

1.2.2 Parameter estimation 13

1.2.3 Relation between Entropy and Likelihood 14

1.3 Reconstruction Methods 14

1.3.1 Directed Binary Configuration Model 15

1.3.2 Directed Enhanced Configuration Model - discrete case 16

1.3.3 Directed Enhanced Configuration Model - continuous case 18

1.3.4 Density corrected Gravity Model 20

2 Generalized inference for efficient reconstruction of weighted networks **23**
2.1 Introduction 24
2.2 Methods 26
2.2.1 Conditional Reconstruction Method (CReM) 26
2.2.2 CReM$_{\mathbf{A}}$ 29
2.2.3 CReM$_{\mathbf{B}}$ 30
2.3 Comparing the two reconstruction methods 32
2.3.1 Likelihood-based comparison 32
2.3.2 Confidence Intervals-based comparison 35
2.4 Discussion 38

3 Entropy-based approach to missing-links prediction **40**
3.1 Introduction 41
3.2 Methods 42
3.3 Data 49
3.4 Results 52
3.5 Discussion 54

4 Two extensions to the missing-link prediction framework **60**
4.1 Introduction 61
4.2 Methods for Weighted Networks 64
4.3 Methods for Binary Bipartite Networks 71
4.4 Results 75
4.5 Discussion 78

5 Conclusion **82**

List of Figures

1.1 Illustration of the network reconstruction problem. Figure adapted from [64]. 2
1.2 Illustration of the link prediction problem. 4
1.3 Examples of different types of graphs. 6
1.4 Example of bipartite graph. 9
1.5 Illustration of the difference between microcanonical and canonical ensembles. 11
1.6 Comparison between the real and estimated weights on a snapshot from the WTW dataset. 22

2.1 Comparison between real observed weights and expected one according to the CReM$_A$ model (blue) and the Gravity Model (red), in a log-log scale. 31
2.2 Likelihood comparison on the WTW and E-mid datasets respectively. Red line indicates CReM$_A$ and blue line CReM$_B$ 33
2.3 Illustration of the procedure for building confidence intervals. 35
2.4 Comparison on the proportion of weights falling into the confidence interval. 36

3.1 Comparison of link-prediction algorithms in the binary undirected case on the WTWdataset. 56

3.2 Comparison of link-prediction algorithms in the binary undirected case on the DIN and E-mid datasets. 57
3.3 Comparison of link-prediction algorithms in the binary directed case on the WTW dataset. 58
3.4 Comparison of link-prediction algorithms in the binary directed case on the DIN dataset. 58
3.5 Comparison of link-prediction algorithms in the binary directed case on the E-mid dataset. 59
3.6 Comparison of link-prediction algorithms in the binary directed case on the foodwebs datasets. 59

4.1 Comparison of link-prediction algorithms in the weighted directed case on the WTW dataset. 75
4.2 Comparison of link-prediction algorithms in the weighted directed case on the E-mid dataset. 76
4.3 Comparison of link-prediction algorithms in the weighted directed case on the WTW and E-mid datasets. 77
4.4 Bipartite link prediction evaluation on Movielens dataset. . 79
4.5 Bipartite link prediction evaluation on Venezuelan Banks dataset. 80

List of Tables

3.1 Network statistics for the WTW, E-mid and DIN datasets. . 50
3.2 Network statistics for the directed version of 18 different food-webs. 51

4.1 Data description of the MovieLens graph, and the Venezuelan Banks and Assets graph 74
4.2 Results: MovieLens performance comparison 78

Acknowledgements

The work presented in this thesis is based on the work by the author and her collaborators, collected in [53, 52, 6]. The work on link and weight prediction in weighted network presented in Chapter 4 is the result of joint work with Tiziano Squartini, and will be the subject of a future paper. Each Chapter of this thesis begins of a statement concerning the origin of the presented work and the relative bibliographic reference.

Vita

July 12, 1990 Born, Chiavari, Italy

2009/2012 **B.SC in Mathematics**
University of Turin, Turin, Italy
Thesis:
First passage time for bivariate diffusion processes with jumps.

2012/2015 **M.SC. in Mathematics**
Curriculum: Probability
University of Turin, Turin, Italy

2013/14 Exchange program:
University of Portsmouth (UK)

2014/15 Research Internship:
International Bureau of Weights and Measures (BIPM), Sevrs, France.
Thesis:
A Kalman Filter model for the atomic clock behaviour: theory and application.
Final Mark: 110/110 cum laude

Publications
and preprint articles

1. F. Parisi, G. Caldarelli, T.Squartini, "Entropy-based approach tomissing-link prediction," in *Applied Network Science*, 3:17, 2018.
2. F. Parisi, T. Squartini, D. Garlaschelli, "A faster horse on a safer trail: generalized inference for efficient reconstruction of weighted networks," *arXiv*, 2018.
3. M. Baltakiene, K. Baltakys, D. Cardamone, F. Parisi, T. Radicioni, M. Torricelli, J. A. van Lidth de Jeude, and F. Saracco, "Maximum entropy approach to link prediction in bipartite networks," *arXiv:1805.04307*, 2018.

Presentations

1. F. Parisi, "Entropy-based approach for missing link imputation," at *Complex Networks*, Lyon, France, 2017.
2. F. Parisi, "Entropy-based approach for missing link prediction," poster at *Netsci*, Paris, France, 2018.
3. F. Parisi, "Entropy-based approach for missing-link prediction," at *Conferece on Complex Systems*, Thessaloniki, Greece, 2018.
4. F. Parisi, "Reconstruction of weighted networks via conditional entropy," at *Conferece on Complex Systems*, Thessaloniki, Greece, 2018.

Abstract

The work presented in this thesis focuses on an issue that very commonly arise when studying a network: missing information. There are many phenomena that can cause such a lack of knowledge, but prior to any attempt at studying the data, it is desirable to have a knowledge of the network at hand that is as complete as possible. Here I will address specifically two types of missing information problems, namely *network reconstruction* and *link prediction*. In the former case, the network structure is hidden, the only information we have access to is the size of the network and some aggregate node-specific quantity. In the context of link prediction we face a different issue: there is a real underlying network that represents the phenomenon we want to study, of which we can only observe an incomplete version where some links are not present. Our goal will be to identify the most likely candidates to be the missing links and, for weighted networks, their intensity. Both problem will be tackled using entropy-based methods, that guarantee the results to be unbiased. The thesis presents advancements on three major fronts. It generalizes the formalism for network reconstruction, proposing a flexible methodology that allows to include any prior topological knowledge and to derive a compatible, unbiased weighted distribution. It proposes a new approach to link prediction, whose key idea is to tune reconstruction models on the accessible portion of network to infer the partially-observed portion, i.e. the most likely missing links. Finally, in the case of weighted prediction, unlike the vast majority of alternative methods, it provides an explicit recipe to estimate the links weights, together with their confidence intervals.

Chapter 1

Introduction

I could have begun by arguing that networks are everywhere and virtually any discipline can benefit from studying how the interconnectedness of the entities of a system affects the overall phenomenology, but I am confident that if you are reading this work you are already aware of this. What I want to do, is to tell you why I think my thesis might interest you. If you just have a practical problem, and a network that needs to be completed before you can work with it, I hope you will find some useful recipes that apply to your case, and we will soon release the corresponding code to make the application of such recipes fast and pain-free.
If you are, like me, interested in theory as well, I believe that the discussions on the differences between reconstruction methods in terms of the implied network generative processes, and the new proposed flexible framework will interest you. In this case, I think you will like the second Chapter in particular. If you are somehow in between those two worlds, and are searching for a principled method to "fill the gaps" in your data, I would suggest to look for answers into Chapters 3 and 4.

The work presented in this thesis focuses on an issue that very commonly arise when approaching the study of a network: missing information. There are many phenomena that can cause such a lack of knowledge: when studying inter-bank loans, for instance, the detailed information

concerning the structure of the network is not available due to privacy regulations. In biological networks, the discovery of connection between entities of the system often involves experiments that are costly, both in economic terms and time-wise. Thus the network structure at any point in time might be only partially available. When dealing with international data, the different countries involved might release data with different timings, thus making it necessary to infer part of the data structure. Moreover, one wish to account for the possibility that the recorded data is partially incorrect, either due to human error or to the nature of the process we wish to study. Regardless of the reason that caused the lack of information, prior to any attempt at studying the data, it is desirable to have a knowledge of the network at hand that is as complete as possible. Here I will address specifically two types of missing information problems, namely *network reconstruction* and *link prediction*.

In the context of **network reconstruction** the network structure is partially or completely hidden, the only information we have access to is the size - i.e. the number of nodes - of the network and some aggregate node-specific measurable quantity. We will refer to this information as *observables* or *constraints*. The first observation we need to make is that there will be many possible network configurations that are compatible with the same set of constraints (think, for example, of all the possible networks of N nodes having L links). Two different ways exist to enforce the aforementioned constraints: *exactly* or *on average*. We will later discuss in detail the differences between these two options. Figure 1.1

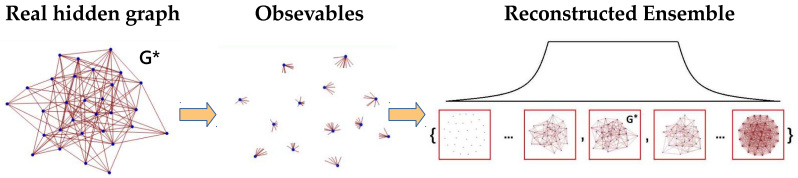

Figure 1.1: Illustration of the network reconstruction problem. Figure adapted from [64].

illustrates the network reconstruction problem in the second case. Notice that the flat part of the distribution over the graph ensemble, in the third panel of the figure, represents the fact that all networks that realize the constraints are assigned the same (maximum) probability, since they are undistinguishable given the available information. Regardless of whether we imposed the constraints strictly or not, once we have assigned a probability to each graph, the expected value of any quantity of interest can be computed by averaging over the compatible configurations [63].
In the context of **link prediction** we face a different issue. Let us assume that there is a real underlying network that correctly represents the phenomenon we want to study, of which we can only observe an incomplete version where some links are not present. Our goal will be to identify the most likely candidates to be the missing links. If we are dealing with weighted networks, i.e. networks whose links have different intensities, the purpose of the link prediction procedure is also to estimate the links weights. See Figure 1.2 for a graphic representation of the link prediction problem, both for binary and weighted networks. Unlike the case of reconstruction, the outcome of the link prediction algorithm will be a single network realization, with the same number of links as the underlying network.

Before starting to introduce the concepts that we will need to build upon in the corpus of the thesis, let me underline the degree of novelty of the work and its main contributions. The thesis presents advancements on three major fronts.

1. It generalizes the formalism for network reconstruction, proposing a flexible methodology that allows to include any prior topological knowledge and to derive a compatible, unbiased weighted distribution[1].

2. On the link prediction front, we propose a new approach, whose

[1] A Python routine of our method will be also shortly released, allowing for a fast and straightforward application of it also by non-experts.

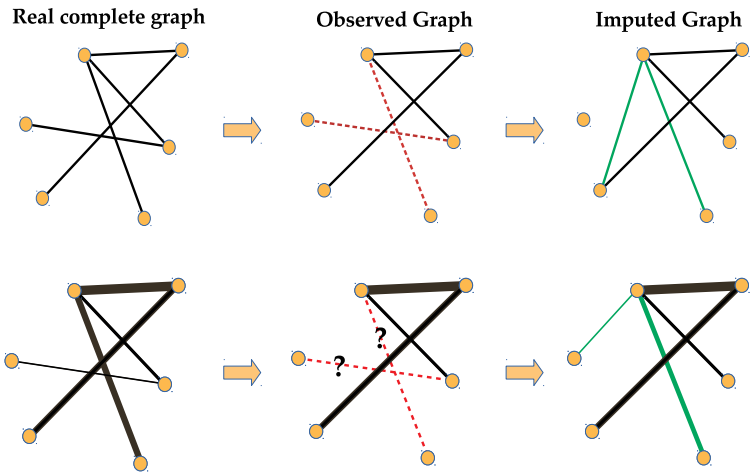

Figure 1.2: Illustration of the link prediction problem, for binary networks (top) and weighted networks (bottom). Some links of the real network are not observed (dashed red lines). The procedure returns the most likely candidates to be the missing links (green lines), and in the weighted case, an estimate of the links weights (represented by the line width).

key idea is to move from the evidence that a network is *reconstructable* to make inference on missing links. That is, we tune reconstruction models on the accessible portion of network to infer the partially-observed portion, i.e. the links most likely to be missing.

3. Finally, in the case of weighted prediction, we provide an explicit recipe to estimate the weights to be attributed to the imputed links, together with confidence intervals for such estimates. Our algorithm substantially enriches the literature on the topic, representing one of the very few methods for weights estimation not resting on any *a posteriori* parameters estimation.

1.1 Basic definitions and notation

We will start from the basic definitions, introduce some notation and present the key concepts upon which this study is based and from which evolves. The interested reader will find references to more comprehensive accounts [11].

Definition 1. A *graph* (or *network*) G = (V, E) is defined by a couple, where V is the set of elements, called *nodes* or vertices and E is the set of *edges*. An edge is a pair of nodes and represents the existence of a link between one node and the other.

We will generally use the term *graph* to indicate the mathematical abstract representation of a system, and the term *network* to refer to a real instance of such structure, derived from some real world phenomenon.

If there is a link between nodes i and j we say that they are *connected*. The *neighbours* of node i are all nodes connected to i.

Definition 2. A graph is *directed* if the edges have a direction associated to them. In this case, an edge becomes an *ordered* pair on nodes, which represents the existence of a link from one node to the other. A *weighted* graph is a graph whose edges are given a *weight*, i.e. a scalar number representing the intensity of the connection.

Figure 1.3 shows the representation of different types of graphs, binary and weighted, directed and undirected.
Each graph admits a standard mathematical representation. Denoting with *N* the number of nodes in the graph we can define

Definition 3. The *adjacency matrix* $\mathbf{A} = \{a_{ij}\}$ of a graph is a $N \times N$ matrix whose rows and columns are associated to the graph nodes. We have

$$a_{ij} = \begin{cases} 1 & \text{there is a link from } i \text{ to } j \\ 0 & \text{otherwise.} \end{cases} \tag{1.1}$$

In the case of weighted networks, the adjacency matrix $\mathbf{W}$ is a $N \times N$ matrix for which w_{ij} is the value of the *link weight* if i and j are connected, and zero otherwise.

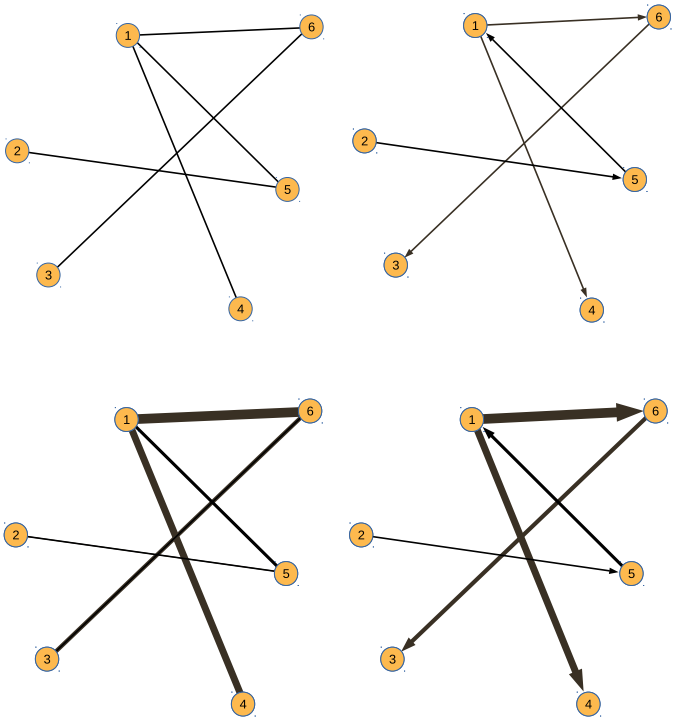

Figure 1.3: Examples of different types of graphs. Top line shows binary graphs, undirected (left) and directed (right). Bottom line shows weighted graphs: undirected (left) and directed (right).

As an example, the adjacency matrices of the toy networks of Figure

1.3 read as

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \quad \mathbf{A}^d = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$
$$\mathbf{W} = \begin{pmatrix} 0 & 0 & 0 & 5 & 2 & 8 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \\ 5 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 0 \\ 8 & 0 & 3 & 0 & 0 & 0 \end{pmatrix} \quad \mathbf{W}^d = \begin{pmatrix} 0 & 0 & 0 & 5 & 0 & 8 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \end{pmatrix}. \tag{1.2}$$

We denoted with apex d the adjacencies relative to directed graphs. We observe that when the network is *undirected*, the adjacency matrix is *symmetric*.

When we deal with a weighted network, the purely topological information is encoded in the *binary* adjacency matrix, defined as $\mathbf{A} = \Theta(\mathbf{W})$, where $\Theta(\cdot)$ is the Heaviside step function. This is a convenient notation, and the step function is applied element by element. In its standard formulation H is not defined in zero, therefore we apply the definition

$$H(x) = \mathbf{1}_{(0,\infty)}(x), \tag{1.3}$$

which implies

$$\Theta(w_{ij}) = \begin{cases} 1 & \text{if } w_{ij} > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{1.4}$$

Once we have this representation of a graph we can use it to express many of the key quantities we are interested in measuring. We start from two very basic ones: the node degree and strength.

Definition 4. In the case of undirected networks, the node *degree* is the number of connections of a node. It can be computed summing all the entries of the binary adjacency matrix of the row or column relative to

the node. For node i we have

$$k_i = \sum_j a_{ij} = \sum_j a_{ji}. \tag{1.5}$$

The *degree sequence* of a graph is the non-increasing sequence of its nodes degrees. In the case of directed networks we define the *in-degree* as the number of incoming links and the *out-degree* as the number of outgoing links of a node. We have

$$k_i^{in} = \sum_j a_{ji} \qquad k_i^{out} = \sum_j a_{ij}. \tag{1.6}$$

In the case of weighted networks we can also define

Definition 5. For undirected networks, the node *strength* is the total flux going through a node, that is the sum of the weights of all the links passing through the node. It can be computed summing all the entries of the *weighted* adjacency matrix of the row or column relative to the node. For node i we have

$$s_i = \sum_j w_{ij} = \sum_j w_{ji}. \tag{1.7}$$

The *strength sequence* of a graph is the non-increasing sequence of the nodes strengths. In the case of directed networks, we define the *in-strength* as the total incoming flux and the *out-strength* as the total outgoing flux of a node. We have

$$s_i^{in} = \sum_j w_{ji} \qquad s_i^{out} = \sum_j w_{ij}. \tag{1.8}$$

1.1.1 Bipartite graphs

So far, we considered graphs where all the nodes played the same role and can connect with all other nodes in the graph. There exist types of graphs for which this is no longer true.

Definition 6. A *bipartite graph* is a graph whose nodes can be divided into two disjoint sets, called **layers**, such that all links connect a node in a layer to a node in the other layer.

We will denote the layers as $\top$ and $\bot$. The nodes of the $\top$ layer will be indicated with Latin letters and the ones from the $\bot$ ones with Greek letters. Figure 1.4 presents an example of a bipartite graph.

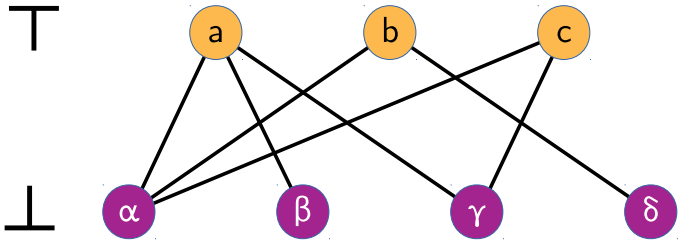

Figure 1.4: Example of bipartite graph.

Definition 7. Given a bipartite graph with $N_\top$ nodes in layer $\top$ and $N_\perp$ in layer $\perp$, the *bipartite adjacency matrix* $\mathbf{M}$ is the $N_1 \times N_2$ matrix such that

$$m_{i\alpha} = \begin{cases} 1 & \text{if } i \text{ and } \alpha \text{ are connected} \\ 0 & \text{otherwise.} \end{cases} \tag{1.9}$$

As an example, the graph in Figure 1.4 has adjacency matrix

$$M = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

Definition 8. The *degree* of a node is given by

$$k_i^\top = \sum_\alpha m_{i\alpha} \qquad k_\alpha^\perp = \sum_i m_{i\alpha} \tag{1.10}$$

depending on the layer it belongs to.

The *degree sequences* of a bipartite graph are the degree sequences of each layer. The degree sequences of the example graph of Figure 1.4 are

$$\vec{k}^{\,\top} = \{3, 2, 2\} \qquad \vec{k}^{\perp} = \{3, 2, 1, 1\}.$$

1.2 Graph Ensembles and the ERG model

When we talk about the *complex* properties of a network we refer to all the phenomena that we observe on the network that are not easily ascribable

to the local properties of the single entities composing the system. Local constraints, like the node degree or strength, are important because they represent a natural way to estimate the node "importance" and therefore to estimate the impact of such factor on the patterns of interconnections. In order to identify these complex phenomena, one will find the need to pose the question: what are the properties of a network that derive directly from its local characteristics?

To answer this question we introduce the concept of a **graph ensemble**. A graph ensemble $\mathfrak{G} = \{\mathbf{G}\}$ is the set of all graphs that satisfy certain constraints. In what follows, the graphs of an ensemble will always have the same number of nodes. There is a choice to be made when building the ensemble. One option is to require all the graphs in the ensemble to satisfy the constraints strictly, e.g. we take all graphs of 4 nodes with degree sequence $\{3, 2, 2, 1\}$. This originates what is called a *microcanonical ensemble*, in analogy with Statistical Physics. Another possibility is to relax the requirement and to only ask that the desired quantities are preserved *on average* over the ensemble. This originates a *(grand) canonical ensemble*. Figure 1.5 illustrates the difference between the two ensemble choices. In the following we will focus on the latter ensemble structure because it is more resilient to error in the imposed constraints. In fact, the microcanonical ensemble assigns null probability to all graphs that have values of constraints that do not exactly correspond to the observed ones. Now, let us assume that instead of having access to the exact constraints, $\vec{C}^*$, calculated on the network we want to reconstruct, $\mathbf{G}^*$, we observed a slightly perturbed version $\vec{C}^p$. If applying the microcanonical ensemble, then $\mathbf{G}^*$ will be assigned zero probability, since the value of the constraints does not coincide with $\vec{C}^p$. In the canonical case, instead, $\mathbf{G}^*$ will have a non-null probability that, for small perturbations of $\vec{C}^*$, will be close to the maximum probability. This characteristic will be fundamental in the context of link prediction, as we shall see.

When building a canonical ensemble of graphs, it will contain all the possible graphs of given number of nodes. It is clear, that not all this graphs are equally likely to give rise to the initial degree sequence. It appears evident that those graphs need to be *weighted* by a probability

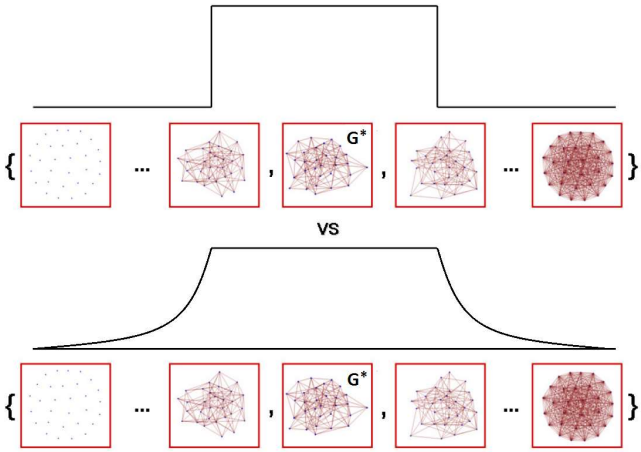

Figure 1.5: Illustration of the difference between microcanonical (above) and canonical (below) ensembles. The former assign non-zero probability only to the graphs that satisfy the constraints exactly, thus inducing an uniform distribution over the admissible configurations, while the latter consider all graphs, assigning maximum (and equal) probability to all the graphs satisfying the constraints, but allows for other graph configurations. Figure from [64].

that represents how likely they are to originate the selected observable. To determine this probability distribution rigorously, we will introduce the Exponential Random Graph framework. For a concise but interesting presentation of the topic the reader can refer to [21].

1.2.1 The Exponential Random Graph model

We start with a network $\mathbf{G}^*$, which we want to model. We select a certain observable $\vec{C}$ to study its impact on the structure of the considered network. We measure the values of the observable on $\mathbf{G}^*$, obtaining $\vec{C}^* = C_1^*, ..., C_r^*$. We build the ensemble $\mathfrak{G}$ of all networks that have the same number of nodes as $\mathbf{G}^*$ and the same types of links (e.g. directed/undirected, binary or weighted). Our goal is to find a distribution

P over $\mathfrak{G}$ such that the expected value of the observable $\vec{C}$, with respect to P, matches a target value, $\vec{C}^t$. (Tipically such target value will correspond to the observed value of the constraints, i.e. $\vec{C}^t = \vec{C}^*$). That is we want to determine P such that

$$\langle C_\alpha \rangle_P = \sum_{\mathbf{G}\in\mathfrak{G}} C_\alpha(\mathbf{G}) P(\mathbf{G}) = C_\alpha^t \qquad \forall \alpha = 1, ..., r. \tag{1.11}$$

The procedure to determine such distribution comes from information theory ans statistical mechanics [36, 37, 17]. This involves the **maximization of the Shannon entropy** of the problem. The rationale of this approach is that maximizing such quantity *under a set of constraints* will allow us to find the distribution, among the ones allowed by the information we have, that is maximally random. This is equivalent to say that we will obtain an *unbiased* distribution, that does not use any assumption on the data except from the information we have from the observables. In the words of Jaynes [37], given a set of constraints, referred to as data,

> [The maximum entropy distribution] in effect creates a model for which those data would have been sufficient statistics.

Therefore, in order to determine the distribution P over $\mathfrak{G}$ that satisfies Eq. 1.11, we need to maximize the quantity

$$S(P) = -\sum_{\mathbf{G}\in\mathfrak{G}} P(\mathbf{G}) \log P(\mathbf{G}), \tag{1.12}$$

under two set of constraints: a simple normalization condition $\sum_{\mathbf{G}\in\mathfrak{G}} P(\mathbf{G}) = 1$, and the condition from (1.11). Therefore the observable we selected will act as the problem constraints, and this explain the subtle idea of indicating such constraints with the letter $\vec{C}$. In order to perform this constrained maximization we express the Lagrangian of the problem as

$$\mathscr{L} = S(P) + \mu \left(1 - \sum_{\mathbf{G}\in\mathfrak{G}} P(\mathbf{G}) \right) + \sum_\alpha \lambda_\alpha \left(C_\alpha^* - \sum_{\mathbf{G}\in\mathfrak{G}} C_\alpha(\mathbf{G}) P(\mathbf{G}) \right) \tag{1.13}$$

Differentiating the Lagrangian with respect to P and setting it equal to zero yields

$$\frac{\partial \mathscr{L}}{\partial P(\mathbf{G})} = -\log P(\mathbf{G}) + 1 - \mu - \sum_{\alpha} \lambda_\alpha C_\alpha(\mathbf{G}) = 0 \tag{1.14}$$

We denote with

$$H(\mathbf{G}) = \sum_{\alpha} \lambda_\alpha C_\alpha(\mathbf{G}), \tag{1.15}$$

the **Hamiltonian** associated with the constraints, and solving for P we obtain

$$P(\mathbf{G}) = \frac{e^{-H(\mathbf{G})}}{Z}, \tag{1.16}$$

where $Z = \sum_{\mathbf{G}} e^{-H(\mathbf{G})}$ is the **partition function** that assures that the normalization condition is satisfied. Equation (1.16) represents the functional form of the distribution over $\mathfrak{G}$ given by the Exponential Random Graph model. The specific functional form of P will be further specified when a choice for the constraints will be made.

1.2.2 Parameter estimation

In the previous Section we have seen how to generate a network ensemble with a probability distribution that ensures that a set of constraints is satisfied on average. However, we have not yet explained how to determine the values of the $\vec{\lambda}$ parameters, that are the Lagrange multipliers associated with the constraints. Our goal is to determine the value $\vec{\lambda}^*$ such that the desired constraints are satisfied. Previous works [25, 63] showed that for the particular class of problems given by the ERG framework, the choice that ensures that the constraints match the target value $\vec{C}^*$ is given by the parameters that maximize the likelihood of the problem. Therefore the parameter estimation becomes a maximum likelihood problem. The parameters $\vec{\lambda}^*$ can therefore be determined as

$$\vec{\lambda}^* = \underset{\vec{\lambda}}{\operatorname{argmax}}\, \mathcal{L}(\vec{\lambda}), \tag{1.17}$$

where

$$\mathcal{L}(\vec{\lambda}) = \log P(\mathbf{G}|\vec{\lambda}). \tag{1.18}$$

1.2.3 Relation between Entropy and Likelihood

We indicated with $\vec{\lambda}^*$ the value of the parameters in (1.13) such that the constraints are satisfied, that is such that

$$\langle \vec{C} \rangle^* = \langle \vec{C} \rangle (\vec{\lambda}^*) = \sum_{\mathbf{G}} \vec{C}(\mathbf{G}) P(\mathbf{G}|\vec{\lambda}^*) = \vec{C}(\mathbf{G}^*) = \vec{C}^*. \tag{1.19}$$

Now, we can rewrite (1.12) as

$$\begin{aligned} S(P) &= -\sum_{\mathbf{G}} P(\mathbf{G})[-H(\mathbf{G}) - \log Z] \\ &= \sum_{\mathbf{G}} P(\mathbf{G}) H(\mathbf{G}) + \log Z \\ &= \langle H \rangle + \log Z \\ &= \langle H \rangle - H(\mathbf{G}) - \log P(\mathbf{G}). \end{aligned} \tag{1.20}$$

When we evaluate this in $\vec{\lambda} \equiv \vec{\lambda}^*$, we obtain

$$S(\vec{\lambda}^*) = S(P_{\vec{\lambda}^*}) = \langle H \rangle (\vec{\lambda}^*) - H^* - \log P(\mathbf{G}|\vec{\lambda}^*) \tag{1.21}$$

Now, using (1.19), we have

$$\langle H \rangle (\vec{\lambda}^*) = \sum_{\alpha} \lambda_\alpha^* \langle C_\alpha \rangle (\vec{\lambda}^*) = \sum_{\alpha} \lambda_\alpha^* C_\alpha^* = H^*. \tag{1.22}$$

Therefore, (1.21) becomes

$$S(\vec{\lambda}^*) = -\log P(\mathbf{G}|\vec{\lambda}^*). \tag{1.23}$$

1.3 Reconstruction Methods

The ERG framework has a natural application for the task of network reconstruction. In this setting we have a network on which we wish to perform some analysis, but that we are unable to observe. The only information we have access to is the size of the network and some aggregate node-specific measurable quantities, which we will call *observables* or *constraints*. In binary networks it typically is the degree sequence and

in weighted ones it typically is the strength sequence and link density for instance. Our objective is to reconstruct the full network given those observables. It is easy to see that this problem can be framed within the ERG model, which will be the starting point of all the methods presented in the following. Each method will be fully determined once the set of observables/constraints is defined. We will now outline a few maximum-entropy methods that have been proposed in the literature for the task of network reconstruction, for binary and weighted networks. We will present the directed version of each method, since the undirected case follows easily as a natural simplification. The first model presented is devoted to the reconstruction of binary networks.

1.3.1 Directed Binary Configuration Model

Following the notation introduced in Section 1.1, we denote (the adjacency matrix of) a binary graph with $\mathbf{A}$. The quantity that we assume to be able to observe from a hidden/unavailable network is the degree sequence. This choice of constraints originates a widely studied class of models called Configuration Model [54]. Since we are dealing with directed networks, we are going to constrain both the in- and out- degree sequences. The model here presented was introduced in [63]. The Hamiltonian of the problem reads

$$H(\mathbf{A}) = \sum_i [\alpha_i k_i^{out}(\mathbf{A}) + \beta_i k_i^{in}(\mathbf{A})] = \sum_{i \neq j} (\alpha_i + \beta_j) a_{ij}. \tag{1.24}$$

As a consequence the partition function becomes

$$Z = \sum_{\mathbf{A}} e^{-H(\mathbf{A})} = \prod_{i \neq j} (1 + e^{-\alpha_i - \beta_j}). \tag{1.25}$$

The overall distribution the reads

$$P(\mathbf{A}) = \prod_{i \neq j} p_{ij}^{a_{ij}} (1 - p_{ij})^{(1 - a_{ij})}, \tag{1.26}$$

with

$$p_{ij} = \frac{x_i y_j}{1 + x_i y_j}, \tag{1.27}$$

where $x_i = e^{-\alpha_i}$ and $y_j = e^{-}\beta_j$. Equation (1.26) tells us that the overall distribution factorizes, that is that the probability of a link being present between two nodes is independent of the presence of other links.
In order to determine the value of the coefficients in (1.26), we maximize the likelihood from (1.18), that in this case reads

$$\mathcal{L}(\vec{x},\vec{y}) = \sum_i [k_i^{out}(\mathbf{A}^*)\log x_i + k_i^{in}(\mathbf{A}^*)\log y_i] - \sum_{i\neq j}\log(1+x_i y_j). \quad (1.28)$$

This implies the conditions,

$$\begin{aligned} \langle k_i^{out}\rangle &= \sum_{j\neq i} p_{ij} = \sum_{j\neq i}\frac{x_i y_j}{1+x_i y_j} = k_i^{*out} \\ \langle k_i^{in}\rangle &= \sum_{j\neq i} p_{ji} = \sum_{j\neq i}\frac{x_j y_i}{1+x_j y_i} = k_i^{*in} \end{aligned} \quad , \quad (1.29)$$

which correspond to imposing the preservation of the constraints.
The complexity of the algorithm is the one of solving a system of $2N$ coupled equations.

1.3.2 Directed Enhanced Configuration Model - discrete case

The extension to the weighted case of the binary configuration model involves the choice of both binary and weighted constraints: the in- and out- degree sequences and in- and out- strength sequences. In fact, it has been shown in the paper that first proposed the undirected version of the model [48], that the information contained in the degree sequence is not reducible to the one contained in the strength sequence. In this case the Shannon entropy to be maximized is expressed as

$$S = -\sum_{\mathbf{W}} Q(\mathbf{W})\log Q(\mathbf{W}), \quad (1.30)$$

relative to the probability distribution over the space of the weighted graph configuration with given number of nodes. The constraints derive from keeping constant, on average, the in- and out- degree and strength

sequences:

$$\begin{aligned}
\langle k_i^{out} \rangle &= \sum_W Q(\mathbf{W}) k_i^{out}(\mathbf{W}) \\
\langle k_i^{in} \rangle &= \sum_{\mathbf{W}} Q(\mathbf{W}) k_i^{in}(\mathbf{W}) \\
\langle s_i^{out} \rangle &= \sum_{\mathbf{W}} Q(\mathbf{W}) s_i^{out}(\mathbf{W}) \\
\langle s_i^{in} \rangle &= \sum_W Q(\mathbf{W}) s_i^{in}(\mathbf{W})
\end{aligned} \tag{1.31}$$

The probability distribution, as in the general ERG framework, takes the functional form

$$Q(\mathbf{W}) = \frac{e^{-H(\mathbf{W})}}{Z}, \tag{1.32}$$

where $Z = \sum_{\mathbf{W}} e^{-H(\mathbf{W})}$ is the partition function and

$$\begin{aligned}
H(\mathbf{W}) =& \sum_{i=1}^{N} \left[\alpha_i^{out} k_i^{out}(\mathbf{W}) + \alpha_i^{in} k_i^{in}(\mathbf{W}) + \right. \\
&\left. + \beta_i^{out} s_i^{out}(\mathbf{W}) + \beta_i^{in} s_i^{in}(\mathbf{W}) \right] \\
=& \sum_{i,j} \left[(\alpha_i^{out} + \alpha_j^{in}) \Theta[w_{ij}] + (\beta_i^{out} + \beta_j^{in}) w_{ij} \right].
\end{aligned}$$

From this we can compute

$$\begin{aligned}
Z &= \sum_{\mathbf{W}} e^{-H(\mathbf{W})} = \prod_{i,j} \left[\sum_{w_{ij}=0}^{\infty} e^{-(\alpha_i^{out} + \alpha_j^{in}) \Theta[w_{ij}] - (\beta_i^{out} + \beta_j^{in}) w_{ij}} \right] \\
&= \prod_{i,j} \left[1 + e^{-(\alpha_i^{out} + \alpha_j^{in})} \sum_{w_{ij}=0}^{\infty} e^{-(\beta_i^{out} + \beta_j^{in}) w_{ij}} \right] = \\
&= \prod_{i,j} \left[1 + \frac{e^{-(\alpha_i^{out} + \alpha_j^{in})}}{e^{\beta_i^{out} + \beta_j^{in}} - 1} \right].
\end{aligned} \tag{1.33}$$

which implies

$$Q(\mathbf{W}) = \prod_{i,j} q_{ij}(w) = \prod_{i,j} \frac{\left(x_i^{out} x_j^{in} \right)^{\Theta[w]} \left(y_i^{out} y_j^{in} \right)^{w} \left(1 - y_i^{out} y_j^{in} \right)}{1 - y_i^{out} y_j^{in} + x_i^{out} x_j^{in} y_i^{out} y_j^{in}}. \tag{1.34}$$

Therefore we have

$$p_{ij} = 1 - q_{ij}(0) = \frac{x_i^{out} x_j^{in} y_i^{out} y_j^{in}}{1 - y_i^{out} y_j^{in} + x_i^{out} x_j^{in} y_i^{out} y_j^{in}} \tag{1.35}$$

$$\langle w_{ij} \rangle = \frac{p_{ij}}{1 - y_i^{out} y_j^{in}}. \tag{1.36}$$

The likelihood function corresponding to (1.18) is given by

$$\begin{aligned}
&\mathcal{L}(\mathbf{W}^*) = \ln Q(\mathbf{W}^*) = \sum_{i,j} \ln q_{ij}(w_{ij}^*) = \\
&= \sum_i \left(k_i^{out*} \ln x_i^{out} + k_i^{in*} \ln x_i^{in} - s_i^{out*} \log y_i^{out} + s_i^{in*} \log y_i^{in} \right) + \\
&- \sum_{i,j} \ln \left(1 - y_i^{out} y_j^{in} + x_i^{out} x_j^{in} y_i^{out} y_j^{in} \right).
\end{aligned} \tag{1.37}$$

Differentiating the likelihood expression with respect to the parameters yields the system of $4N$ interdependent equations

$$\begin{aligned}
\langle k_i^{out} \rangle &= \sum_{j \neq i} p_{ij} = \sum_{j \neq i} \frac{x_i^{out} x_j^{in} y_i^{out} y_j^{in}}{1 - y_i^{out} y_j^{in} + x_i^{out} x_j^{in} y_i^{out} y_j^{in}} = k_i^{*out} \\
\langle k_i^{in} \rangle &= \sum_{j \neq i} p_{ji} = \sum_{j \neq i} \frac{x_j^{out} x_i^{in} y_j^{out} y_i^{in}}{1 - y_j^{out} y_i^{in} + x_j^{out} x_i^{in} y_j^{out} y_i^{in}} = k_i^{*in} \\
\langle s_i^{out} \rangle &= \sum_{j \neq i} \langle w_{ij} \rangle = \sum_{j \neq i} \frac{p_{ij}}{1 - y_i^{out} y_j^{in}} = s_i^{*out} \\
\langle k_i^{in} \rangle &= \sum_{j \neq i} \langle w_{ij} \rangle = \sum_{j \neq i} \frac{p_{ji}}{1 - y_j^{out} y_i^{in}} = s_i^{*in}.
\end{aligned} \tag{1.38}$$

1.3.3 Directed Enhanced Configuration Model - continuous case

In this Section we extend the model presented in the previous section by assuming the link weights to be continuous. [2]
In this case the Shannon entropy to be maximized is expressed as

$$S = -\int_W Q(\mathbf{W}) \log Q(\mathbf{W}), \tag{1.39}$$

[2]We are aware that a paper treating the continuous case is currently in progress [15], however the extension here presented was developed independently.

relative to the probability distribution over the space of the weighted graph configuration with given number of nodes. The constraints derive from keeping constant, on average, the in- and out-degree and strength sequences:

$$\begin{aligned}
\langle k_i^{out} \rangle &= \int_{\mathbf{W}} Q(\mathbf{W}) k_i^{out}(\mathbf{W}) \\
\langle k_i^{in} \rangle &= \int_{\mathbf{W}} Q(\mathbf{W}) k_i^{in}(\mathbf{W}) \\
\langle s_i^{out} \rangle &= \int_{\mathbf{W}} Q(\mathbf{W}) s_i^{out}(\mathbf{W}) \\
\langle s_i^{in} \rangle &= \int_{\mathbf{W}} Q(\mathbf{W}) s_i^{in}(\mathbf{W})
\end{aligned} \tag{1.40}$$

The probability distribution, as in the general Exponential Random Graph framework [54], takes the functional form

$$Q(\mathbf{W}) = \frac{e^{-H(\mathbf{W})}}{Z}, \tag{1.41}$$

where $Z = \int_{\mathbf{W}} e^{-H(\mathbf{W})}$ is the partition function and

$$\begin{aligned}
H(\mathbf{W}) = &\sum_{i=1}^{N} \left[\alpha_i^{out} k_i^{out}(\mathbf{W}) + \alpha_i^{in} k_i^{in}(\mathbf{W}) + \right. \\
&\left. + \beta_i^{out} s_i^{out}(\mathbf{W}) + \beta_i^{in} s_i^{in}(\mathbf{W}) \right] \\
= &\sum_{i,j} \left[(\alpha_i^{out} + \alpha_j^{in}) \Theta[w_{ij}] + (\beta_i^{out} + \beta_j^{in}) w_{ij} \right]
\end{aligned}$$

From this we can compute

$$\begin{aligned}
Z &= \int_{\mathsf{W}} e^{-H(\mathsf{W})} = \\
&= \prod_{i,j} \int_0^{\infty} \left[\delta_{w_{ij},0} + \Theta(w_{ij}) \right] e^{-(\alpha_i^{out} + \alpha_j^{in}) \Theta[w_{ij}] - (\beta_i^{out} + \beta_j^{in}) w_{ij}} \, dw_{ij} = \\
&= \prod_{i,j} \left[1 + e^{-(\alpha_i^{out} + \alpha_j^{in})} \int_0^{\infty} e^{-(\beta_i^{out} + \beta_j^{in}) w_{ij}} \, dw_{ij} \right] = \\
&= \prod_{i,j} \left[1 + \frac{e^{-(\alpha_i^{out} + \alpha_j^{in})}}{\beta_i^{out} + \beta_j^{in}} \right].
\end{aligned} \tag{1.42}$$

which implies

$$Q(\mathbf{W}) = \prod_{i,j} q_{ij}(w) = \prod_{i,j} \frac{\left(x_i^{out} x_j^{in}\right)^{\Theta[w]} e^{-(\beta_i^{out}+\beta_j^{in})w}}{1 + x_i^{out} x_j^{in}/(\beta_i^{out}+\beta_j^{in})}, \tag{1.43}$$

from which we have

$$p_{ij} = 1 - q_{ij}(0) = \left[1 + (\beta_i^{out} + \beta_j^{in})/(x_i^{out} x_j^{in})\right]^{-1}, \tag{1.44}$$

$$\langle w_{ij} \rangle = \frac{p_{ij}}{\beta_i^{out} + \beta_j^{in}} \tag{1.45}$$

The likelihood function to maximize to estimate the model parameter reads

$$\begin{aligned}
&\mathcal{L}(\mathbf{W}^*) = \ln Q(\mathbf{W}^*) = \sum_{i,j} \ln q_{ij}(w_{ij}^*) = \\
&= \sum_i \left(k_i^{out*} \ln x_i^{out} + k_i^{in*} \ln x_i^{in} - s_i^{out*} \beta_i^{out} - s_i^{in*} \beta_i^{in} \right) - \\
&- \sum_{i,j} \ln \left(1 + \frac{x_i^{out} x_j^{in}}{\beta_i^{out} + \beta_j^{in}} \right).
\end{aligned} \tag{1.46}$$

Differentiating the likelihood expression with respect to the parameters yields the system of $4N$ interdependent equations

$$\begin{aligned}
\langle k_i^{out} \rangle &= \sum_{j \neq i} p_{ij} = \sum_{j \neq i} \left[1 + (\beta_i^{out} + \beta_j^{in})/(x_i^{out} x_j^{in})\right]^{-1} = k_i^{*out} \\
\langle k_i^{in} \rangle &= \sum_{j \neq i} p_{ji} = \sum_{j \neq i} \left[1 + (\beta_j^{out} + \beta_i^{in})/(x_j^{out} x_i^{in})\right]^{-1} = k_i^{*in} \\
\langle s_i^{out} \rangle &= \sum_{j \neq i} \langle w_{ij} \rangle = \sum_{j \neq i} \frac{p_{ij}}{\beta_i^{out} + \beta_j^{in}} = s_i^{*out} \\
\langle k_i^{in} \rangle &= \sum_{j \neq i} \langle w_{ij} \rangle = \sum_{j \neq i} \frac{p_{ij}}{\beta_j^{out} + \beta_i^{in}} = s_i^{*in}.
\end{aligned} \tag{1.47}$$

1.3.4 Density corrected Gravity Model

The methods presented so far all assume the knowledge of the degree sequences of the network to reconstruct. However, there are cases in which

this information is not available. For instance, in the case of interbank loans networks, we typically have access only to the strength sequences and the total density. In this case, a different reconstruction method has been proposed [14], which takes the name of density-corrected Gravity Model (dcGM). This method relies on the heuristic observation that the node in- and out- strengths are correlated with the Lagrange multiplier induced by the in- and out-degrees, that is the x_i and y_j parameter from (1.27). From this, we can write $x_i^{out} = \sqrt{a}s_i^{out}$ and $x_i^{out} = \sqrt{b}s_i^{in}$, that implies, for $z = \sqrt{ab}$,

$$p_{ij} = \frac{z s_i^{out} s_j^{in}}{1 + z s_i^{out} s_j^{in}} \tag{1.48}$$

The only parameter to be determined is z. To do so, we impose the preservation of the total network density $\langle L \rangle$. This gives

$$\langle L \rangle = \sum_i p_{ij} = \sum_i \frac{z s_i^{out} s_j^{in}}{1 + z s_i^{out} s_j^{in}} = L \tag{1.49}$$

This determines the linkage probabilities. As for the link weights, the model imposes the expected weight

$$\langle w_{ij} \rangle = \frac{s_i^{out} s_j^{in}}{W_{tot}}. \tag{1.50}$$

This expression is common to several different models, for instance the MaxEnt algorithm [71, 70]. Empirically, it has shown good agreement with the data, see Figure 1.6 for a comparison between real and expected weights on a snapshot of the World Trade Web dataset. This model has been extended to the case in which we only have knowledge on the density of a subset of the graph [66]. In the same work, the authors also propose a solution for an undesirable property of the dcGM: it does not guarantee the preservation of strengths exactly, due to the need of considering diagonal terms of the form $\langle w_{ii} \rangle$ to ensure that strengths are correctly replicated. The reader can find the details in [66].
In order to have the expected value of the weights equal to (1.50), the

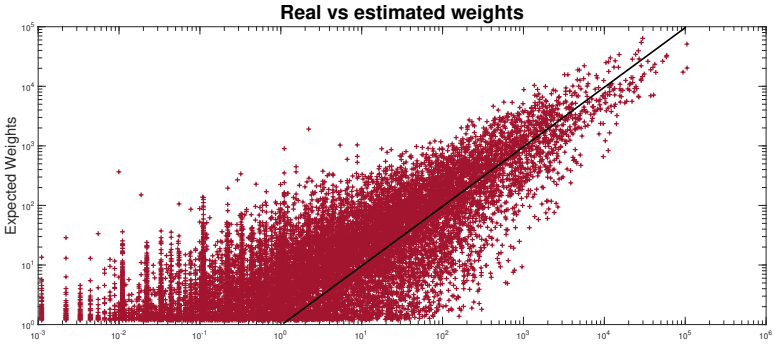

Figure 1.6: Comparison between the real weights and the estimates from (1.50) on a snapshot from the World Trade Web dataset. The plot is in log-log scale and the diagonal line is shown for reference.

model uses a Bernoullian recipe

$$w_{ij} = \begin{cases} \frac{s_i^{out} s_j^{in}}{W_{tot} p_{ij}} & \text{with probability } p_{ij} \\ 0 & \text{otherwise} \end{cases}, \tag{1.51}$$

that is, a link is drawn with probability p_{ij} from (1.48), and its weight is set to $\frac{s_i^{out} s_j^{in}}{W_{tot} p_{ij}}$. This characteristic of the model implies that, when assuming continuous weights, the likelihood of reproducing any real world network is zero.

Chapter 2

Generalized inference for efficient reconstruction of weighted networks

This Chapter is based on the work [53], currently under submission, by F. Parisi, T. Squartini and D. Garlaschelli.

Abstract

Addressing the network reconstruction problem means facing the double challenge represented by the estimation of topology and of link weights. Among the methods proposed so far, some attribute weights on a binary configuration using a completely separate methodology, while others assume that the binary and weighted constraints jointly determine the final configuration. Here we develop a flexible framework that allows to include any prior topological knowledge and to derive a compatible unbiased weighted distribution. Our formulation clearly points out that to derive such a distribution

the knowledge of the *whole* binary distribution is necessary, as opposed to a single binary realization. In fact, either if only one configuration is considered or weights are added deterministically, the likelihood of observing any real network is always zero (assuming continuous weights): therefore, our method has an a priori advantage with respect to the alternative ones, returning a non-zero likelihood for the actual configuration. To this aim, we extend the classical model selection framework, by proposing a generalized likelihood functional that takes as input conditional probability distributions. Another key point of our method is the final model complexity: we propose two specifications of our conditional framework whose numerical implementation is greatly simplified with respect to analogous algorithms.

2.1 Introduction

Network reconstruction is an active field of research within the broader field of complex networks [67, 16]. Addressing the network reconstruction problem means facing the double challenge represented by the estimation of topology and link weights. The task at end consists in determining both binary and weighted distributions, and to understand the interplay between them. Among the methods proposed so far, some assume that the binary and weighted constraints jointly determine the final configuration in terms of both topology and weights while others attribute weights to the binary configuration using a completely separate methodology [3, 30]. Amidst the former ones, a special mention is deserved by the Enhanced Configuration Model [48] . This is defined by simultaneously constraining the degrees and the strengths of nodes which jointly affect the estimation of the two sets of quantities, the linkage probabilities and the weight estimates. Since these are jointly determined on the basis of the same information (i.e. constraints), this implies the impossibility to include purely topological additional information. Examples of algorithms belonging to the second group are those iteratively

adjusting the link weights (e.g. via the RAS recipe [5]) on top of some previously-determined topological structure, in such a way to satisfy the constraints concerning strengths a posteriori. This approach has encountered critiques in [22]. It is important to notice that this kind of procedure assigns weights deterministically, and therefore the likelihood of observing any real matrix is exactly zero, assuming continuous weights. There exist also two-steps algorithms [13] that attempt at overcome the lack of binary information for the topological estimation. However, they have an elevated complexity (the second step requires the solution of the ECM) and the two step procedure is only heuristically motivated. Here we develop a theoretical framework that provides an analytical, unbiased procedure to estimate the weighted structure of a network, once its topology has been determined, thus extending the Exponential Random Graph framework to deal with cases that appeared tractable only via heuristic approaches. The prior information on the topological structure (either already available or obtained using a method of choice) and a number of weighted constraints are taken as input. Subsequently, we derive the unbiased link weight probability. The latter is determined by the maximization of the key quantity of our approach, i.e. the conditional entropy of the weighted distribution given the binary one.

As it will be proven, once the weights are treated as continuous random variables, their distribution, conditional on the existence of a link, is exponential. This consideration also allows to determine confidence intervals for the weights estimates. Another desirable property of the model, in its second specification, is its computational simplicity, as it does not require the numerical solution of several coupled non-linear equations.

We indicate the binary adjacency matrix of the network as $\mathbf{A}$, and we assume that it is a realization from a random variable $\mathcal{A}$. Analogously, the weighted adjacency matrix associated with the graph, $\mathbf{W}$, comes from a random variable $\mathcal{W}$. The probability mass function of the event $\{\mathcal{A} = \mathbf{A}\}$ is denoted with $P(\mathbf{A})$, while $Q(\mathbf{W}|\mathbf{A})$ indicates the conditional multivariate probability density function of $\mathcal{W}$ belonging to a neighbourhood of $\mathbf{W}$, given $\mathcal{A} = \mathbf{A}$. We denote the entries of the weighted adjacency matrix, $\mathbf{W}$, as w_{ij} and their binary counterparts as $a_{ij} = \Theta(w_{ij})$, where

Θ is the Heaviside step function defined in (1.3).

The chapter is organized as follows: in Section 2.2 the proposed Conditional Reconstruction Method (CReM) is analytically derived in the general case, with subsections 2.2.2 and 2.2.3 containing two different specifications of the model deriving from different constraint choices. Section 2.3 presents two alternative methods for model comparison. We apply them to test the relative performance of the two specifications of the CReM and the DECM model (See Section 1.3.3) on different real-world datasets. Finally, in 2.4 we comment on the results and presents the final remarks.

2.2 Methods

2.2.1 Conditional Reconstruction Method (CReM)

Input The procedure takes as input $P(\mathbf{A})$, the distribution over the space of the binary configurations. This can be available as prior (probabilistic) information or computed using any available method. Possible choices for such distribution include for instance the recipes from the DBCM (see Section 1.3.1) or the dcGM (see Section 1.3.4), however it is important to notice that the following results are valid for any choice of $P(\mathbf{A})$. Moreover, the CReM requires a set of weighted constraints $\vec{C}(\mathbf{W})$ to be imposed when deriving the weighted distribution. The unconditional ensemble average of such quantities will correspond to the desired target value. This can be chosen to be an observed quantity or any other value of choice.

Output The goal of this second step is to derive the distribution of the weighted configurations conditional on the binary one. To achieve this, we maximize over $Q(\mathbf{W}|\mathbf{A})$ the *conditional entropy* [17]

$$S(\mathcal{W}|\mathcal{A}) = -\sum_{\mathbf{A}} P(\mathbf{A}) \int_{\mathbf{W_A}} Q(\mathbf{W}|\mathbf{A}) \log Q(\mathbf{W}|\mathbf{A}) d\mathbf{W}. \tag{2.1}$$

We assume that the weights are continuous. We stress that the distribution $Q(\mathbf{W}|\mathbf{A})$ determined by the second step is relative only to the $\mathbf{W}$

compatible with $\mathbf{A}$, that is such that $\Theta(\mathbf{W}) = \mathbf{A}$. We will use the compact notation $\mathbf{W_A}$ to indicate all the $\mathbf{W}$ s.t. $\Theta(\mathbf{W}) = \mathbf{A}$. For all $\mathbf{W}' \notin \mathbf{W_A}$ we have by definition $Q(\mathbf{W}'|\mathbf{A}) \equiv 0$.

Our goal is to maximize, with respect to $Q(\mathbf{W}|\mathbf{A})$, the expression (2.1) under a set of constraints:

1. $\int_{\mathbf{W_A}} Q(\mathbf{W}|\mathbf{A}) d\mathbf{W} = 1 \quad \forall \mathbf{A}$
2. $\sum_{\mathbf{A}} \int_{\mathbf{W_A}} P(\mathbf{A}) Q(\mathbf{W}|\mathbf{A}) C_\alpha(\mathbf{W}) d\mathbf{W} = C^*_\alpha, \forall \alpha.$

The first equation imposes the normalization of the conditional probabilities: since also $P(\mathbf{A})$ is normalized, notice that condition 1. implies $\int_{\mathbf{W}} Q(\mathbf{W}) d\mathbf{W} = 1$. In the second equation $\vec{C}(\mathbf{W}) = \{C_\alpha(\mathbf{W})\}_\alpha$ represents the desired set of constraints taken as input.
The problem Lagrangian can be written as

$$\begin{aligned}\mathscr{L} = &- \sum_{\mathbf{A}} \int_{\mathbf{W}_A} P(\mathbf{A}) Q(\mathbf{W}|\mathbf{A}) \log Q(\mathbf{W}|\mathbf{A}) d\mathbf{W} + \\ &+ \sum_{\mathbf{A}} \mu(\mathbf{A}) \left(1 - \int_{\mathbf{W_A}} Q(\mathbf{W}|\mathbf{A}) d\mathbf{W}\right) + \\ &+ \sum_{\alpha} \lambda_\alpha \left(C^*_\alpha - \sum_{\mathbf{A}} \int_{\mathbf{W}_A} P(\mathbf{A}) Q(\mathbf{W}|\mathbf{A}) C_\alpha(\mathbf{W}) d\mathbf{W}\right).\end{aligned} \tag{2.2}$$

Differentiating this expression with respect to $Q(\mathbf{W}|\mathbf{A})$ and equating to zero, we obtain

$$Q(\mathbf{W}|\mathbf{A}) = \frac{e^{-\sum_\alpha \lambda_\alpha C_\alpha(\mathbf{W})}}{\int_{\mathbf{W_A}} e^{-\sum_\alpha \lambda_\alpha C_\alpha(\mathbf{W})} d\mathbf{W}} = \frac{e^{-H(\mathbf{W})}}{Z_{\mathbf{A}}}, \tag{2.3}$$

where $H(\mathbf{W}) = \sum_\alpha \lambda_\alpha C_\alpha(\mathbf{W})$ is the Hamiltonian and the parameter μ guarantees that condition 1. is satisfied. The final form for the conditional distribution of the weighted configuration is obtained choosing which constraints we wish to impose.

Functional form of likelihood function In alignment with previous results on maximum entropy methods [25, 63, 65], we wish to formulate our problem in terms of maximum likelihood estimation. That is, we

want to be able to re-state the problem is such a form that allows to determine the values of the Lagrange multipliers $\vec{\lambda}$ of (2.2) by solving a maximum likelihood problem. See Section 1.2.3 for the standard case.

Let us consider the integral term of the conditional entropy defined in (2.1). Using (2.3) we can write

$$\begin{aligned} &\int_{\mathbf{W_A}} Q(\mathbf{W}|\mathbf{A}) \log Q(\mathbf{W}|\mathbf{A}) d\mathbf{W} = \\ &\int_{\mathbf{W_A}} Q(\mathbf{W}|\mathbf{A})[-H(\mathbf{W}) - \log Z_{\mathbf{A}}] = \langle H \rangle_{\mathbf{A}} - \log Z_{\mathbf{A}}. \end{aligned} \tag{2.4}$$

Let us now define $\vec{\lambda}^*$ as the value of the parameters for which the constraints are realized, that is

$$\langle \vec{C} \rangle (\vec{\lambda}^*) = \sum_{\mathbf{A}} P(\mathbf{A}) \int_{\mathbf{W_A}} \vec{C}(\mathbf{W}) Q(\mathbf{W}|A, \vec{\lambda}^*) = \vec{C}^*. \tag{2.5}$$

When we evaluate the conditional entropy for such parameter choice, since for both method specifications H is linear in w_{ij}, we obtain

$$\begin{aligned} S(\vec{\lambda}^*) &= -\sum_{\mathbf{A}} P(\mathbf{A})[\langle H \rangle_{\mathbf{A}}(\vec{\lambda}^*) - \log Z_{\mathbf{A}}(\vec{\lambda}^*)] \\ &= -\sum_{\mathbf{A}} P(\mathbf{A}) \log Q(\langle \mathbf{W} \rangle^* | \mathbf{A}), \end{aligned} \tag{2.6}$$

where $\langle \mathbf{W} \rangle^*$ indicates the unconditional ensemble average of $\mathbf{W}$ when the desired constraints are satisfied.

The last equation conveys meaningful information: we started from the expression of the conditional entropy which averages both with respect to the binary configuration and to the weighted one. Imposing certain constraints to be satisfied on average, we determine the value of the vector of parameters $\vec{\lambda}^*$. When (2.1) is evaluated in $\vec{\lambda}^*$, it does no longer contain the averaging with respect to weighted configurations, but instead a single term, $\log Q(\langle \mathbf{W} \rangle^* | \mathbf{A})$, that is then averaged over the space of binary configurations.

Therefore, the functional form of the likelihood function that preserves the dual relation between entropy and likelihood that we have in the

standard case is given by the *generalized likelihood*

$$\mathcal{G}(\vec{\lambda}) = \sum_{\mathbf{A}} P(\mathbf{A}) \log Q(\langle \mathbf{W} \rangle | \mathbf{A}). \tag{2.7}$$

This correspondence between entropy and likelihood has another important consequence: the entropy, when evaluated in $\vec{\lambda}^*$, itself becomes a measure of goodness of fit of the model. Let us observe that the estimation of parameter is a more general problem, arising for instance in the case of parameters estimation for the generative model of a known system, as treated in [**almog2015gdp**].
We are now going to derive the full specification of the model for two different constraint choices: the strength sequences (CReM$_A$) and the conditional weights (CReM$_B$).

2.2.2 CReM$_\mathbf{A}$

For each node i the in- and out- degree and in- and out- strength can be expressed as $k_i^{out} = \sum_j \Theta(w_{ij})$, $k_i^{in} = \sum_j \Theta(w_{ji})$, $s_i^{out} \sum_j w_{ij}$ and $s_i^{in} = \sum_j w_{ji}$. Imposing the preservation of the strength sequences results in an Hamiltonian function of the form:

$$H(\mathbf{W}) = \sum_i (\beta_i^{out} s_i^{out} + \beta_i^{in} s_i^{in}) = \sum_{i \neq j} (\beta_i^{out} + \beta_j^{in}) w_{ij}. \tag{2.8}$$

The partition function can be expressed as

$$Z_A = \prod_{i \neq j} \left[\int_0^\infty e^{-(\beta_i^{out} + \beta_j^{in}) w_{ij}} dw_{ij} \right]^{a_{ij}} = \prod_{i \neq j} \left(\frac{1}{\beta_i^{out} + \beta_j^{in}} \right)^{a_{ij}}. \tag{2.9}$$

Finally, using expression (2.3), we can write

$$Q(\mathbf{W}|\mathbf{A}) = \prod_i e^{-(\beta_i^{out} s_i^{out} + \beta_i^{in} s_i^{in})} \prod_{i \neq j} (\beta_i^{out} + \beta_j^{in})^{a_{ij}}, \tag{2.10}$$

which implies

$$q_{ij}(w|a_{ij}) = (\beta_i^{out} + \beta_j^{in})^{a_{ij}} e^{-(\beta_i^{out} + \beta_j^{in}) w} \tag{2.11}$$

$$q_{ij}(w|a_{ij}=1) = (\beta_i^{out} + \beta_j^{in})e^{-(\beta_i^{out}+\beta_j^{in})w} \tag{2.12}$$

that shows that the weight distribution, conditional on the existence of a link, follows an exponential distribution of parameter $\beta_i^{out} + \beta_j^{in}$.

In order to determine the values of the vector of parameters $\boldsymbol{\beta}^{out}$ and $\boldsymbol{\beta}^{in}$, we maximize the expression of the conditional likelihood, that, substituting the results just derived into (2.7), reads as follows

$$\mathcal{G}_A = -\sum_i (s_i^{out*}\beta_i^{out} + s_i^{in*}\beta_i^{in}) + \sum_{i\neq j} f_{ij}\log(\beta_i^{out} + \beta_j^{in}) \tag{2.13}$$

The expression $f_{ij} = \sum_{\mathbf{A}} P(\mathbf{A})a_{ij}$ represents the value of a_{ij} averaged over the ensemble of binary configurations. This general formulation implies no assumption on the structure of link interdependencies given by $P(\mathbf{A})$. For instance, in the case of the micro-canonical ensemble, where the degree are constrained sharply and not on average on the ensemble, $P(\mathbf{A})$ is constant on all graphs with the same degree sequence. When $P(\mathbf{A})$ factorizes f_{ij} corresponds to the quantity denoted with p_{ij} in the literature [67], that is the linkage probability, which is *independent* from link to link.

Differentiating (2.13) with respect to β_i^{out} and β_i^{in}, $\forall i$, yields the system of 2N coupled equations

$$\begin{cases} s_i^{out} = \sum_j \frac{f_{ij}}{\beta_i^{out}+\beta_j^{in}} \\ s_i^{in} = \sum_j \frac{f_{ij}}{\beta_i^{out}+\beta_j^{in}} \end{cases} \Rightarrow \langle w_{ij}\rangle = \frac{f_{ij}}{\beta_i^{out}+\beta_j^{in}}. \tag{2.14}$$

where f_{ij} is taken as given and therefore excluded from the estimation procedure.

2.2.3 CReM$_{\mathbf{B}}$

In Figure 2.1, we can see, in blue circles, the comparison between the weight estimates delivered from the CReM$_A$ model versus the real weights in blue. The red crosses represent the same comparison but with respect to the weight estimates taken from the density-corrected Gravity Model [14]

$$w_{ij} = \frac{s_i^{out}s_j^{in}}{W_{tot}}, \tag{2.15}$$

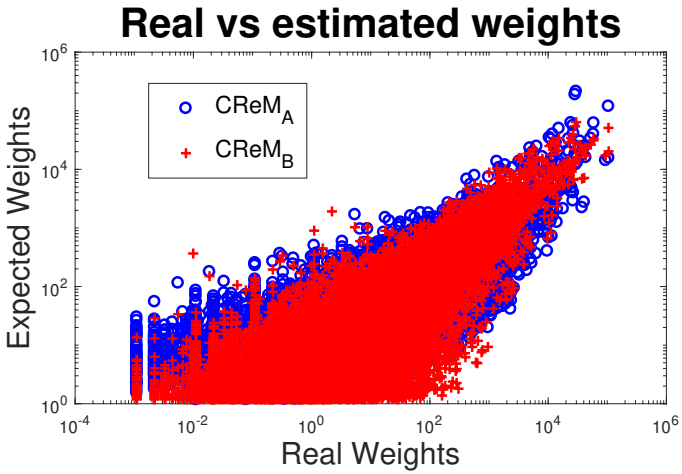

Figure 2.1: Comparison between real observed weights and expected one according to the CReM$_A$ model (blue) and the Gravity Model (red), in a log-log scale.

where $W_{tot} = \sum_i s_i^{out} = \sum_i s_i^{in}$ is the total weight of the network. We can see that the latter estimates shows a better agreement with the data. Therefore, we wish to find a different model specification that allows us to introduce such a functional form for the weight estimates. To do so, in the conditional entropy maximization we formally constrain the values of the weights. In this case, the Hamiltonian reads

$$H(\mathbf{W}) = \sum_{i \neq j} \beta_{ij} w_{ij}. \tag{2.16}$$

The derivation is analogous to the previous case. Given (2.3), the conditional weight distribution will be

$$Q(\mathbf{W}|\mathbf{A}) = \prod_{i \neq j} \frac{e^{-\beta_{ij} w_{ij}}}{\left(\frac{1}{\beta_{ij}}\right)^{a_{ij}}} = \prod_{i \neq j} e^{-\beta_{ij} w_{ij}} (\beta_{ij})^{a_{ij}}. \tag{2.17}$$

As a consequence, also in this case the conditional weight distribution follows an exponential distribution of parameter β_{ij}, since

$$q_i j(w_{ij}|a_{ij} = 1) = \beta_{ij} e^{-\beta_{ij} w_{ij}}. \tag{2.18}$$

The generalized likelihood function, given (2.7), can be expressed as

$$\begin{aligned}\mathcal{G}_B &= \sum_A P(\mathbf{A}) \left[-\sum_{i\neq j} \left(w_{ij}^* \beta_{ij} + a_{ij} \log \beta_{ij} \right) \right] \\ &= \sum_{i\neq j} \left(-w_{ij}^* \beta_{ij} + f_{ij} \log \beta_{ij} \right).\end{aligned} \tag{2.19}$$

Differentiating with respect to β_{ij} leads to the equation

$$w_{ij}^* = \frac{f_{ij}}{\beta_{ij}}. \tag{2.20}$$

Clearly, we cannot really observe the link weights (or there would be no need for reconstruction), but we formally imposed such constraints in order to be able to use the weight estimates from (2.15), which shows good agreement with the data. Such a choice uses as input only the strength sequences of the graph. As a consequence the sufficient statistics for CReM$_A$ and CReM$_B$ are the same: the strength sequences.
If we equate expressions (2.15) and (2.20) we see that to determine the matrix of coefficients $\boldsymbol{\beta}$ we only have to compute

$$\beta_{ij} = \frac{W_{tot}\ f_{ij}}{s_i^{out} s_j^{in}}. \tag{2.21}$$

2.3 Comparing the two reconstruction methods

In this section we will examine the relative performances of the two specifications of CReM and the DECM model (in its continuous specification, as presented in Section 1.3.3). We will propose two different techniques to carry out the comparison, and then comment on the results.

2.3.1 Likelihood-based comparison

The first possible approach is to compare the models through their (generalized) likelihood $\mathcal{G}$. Given any observed strength sequences (or strength and degree for the DECM) we solve the constrained entropy maximization. In particular, we solve the likelihood equations characterizing the

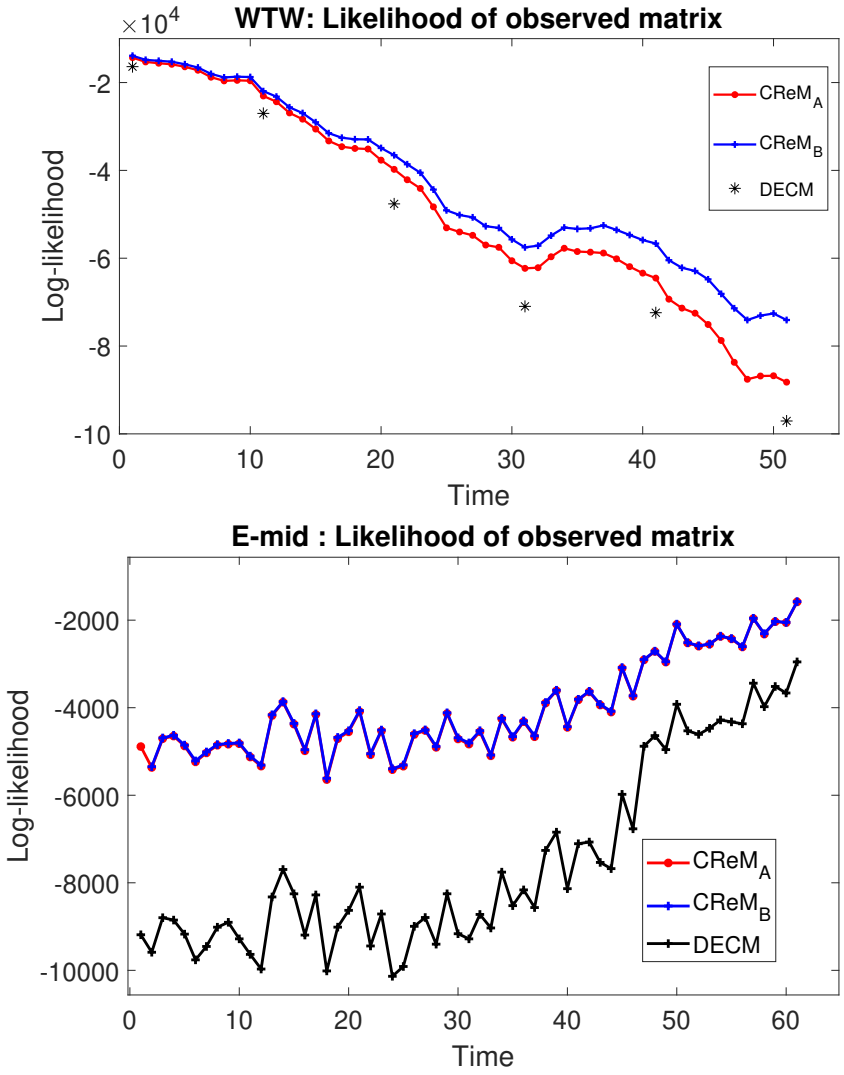

Figure 2.2: Likelihood comparison on the WTW and E-mid datasets respectively. Red line indicates CReM$_A$ and blue line CReM$_B$.

two models, i.e. for CReM$_A$ we determine the vectors β^{out} and β^{in} by

solving the system of equations (2.14), for CReM$_B$, we compute the quantities (2.21) and for the DECM we determine the parameters values solving the system (1.47). For both models A and B, we used as prior binary distribution the linkage probabilities from the dcGM model (See (1.48) in the Section 1.3.4 for details), since methods using such probability form have won several independent horse races [**anand2018missing**, **mazzarisi2017methods**, **ramadiah2017reconstructing**].

Once we have obtained those parameters, we can compute the likelihood of observing the original network by reconstructing it through the different models. In Figure 2.2 we can see the comparison of the values of the (generalized) likelihood relative to models CReM$_A$, CReM$_B$ and DECM, on several snapshots of two different real world datasets, the World Trade Web and the E-mid [33]. Since $\mathcal{G}$ is the (weighted average of) the logarithm of a probability, takes negative values. The closer its value is to zero, the better the model explains the data. Therefore, when using the likelihood to compare two models, we shall prefer the model with the highest values of $\mathcal{G}$. As we can see from the figure, the CReM$_B$ performs better in the WTW dataset, while the performances are extremely similar on the E-mid one.

Generalized AIC test A very common test for model selection is the Akaike Information Criterion (AIC) [2]. For a given set of data, for a generic model m, the standard AIC recipe, reads $\text{AIC}_m = 2k_m - 2\mathcal{L}_m$ can be generalized to the use of a generalized likelihood $\mathcal{G}$ as

$$\text{GAIC}_m = 2k_m - 2\mathcal{G}_m \tag{2.22}$$

where k is the number of estimated parameters of the model and $\mathcal{G}$ the value of the model generalized likelihood computed on the observed data. The number of parameters estimated by each model is $k_{DECM} = 4N$, $k_{CReM_A} = 2N + 1$, and $k_{CReM_B} = 2N + 1$.

The additional parameter comes from constraining the link density, as with the dcGM recipe (see Section 1.3.4 for details). For both CReM$_A$ and CReM$_B$ the sufficient statistics are the vectors of in and out strengths and the total density of links. Given this observation, we note that the

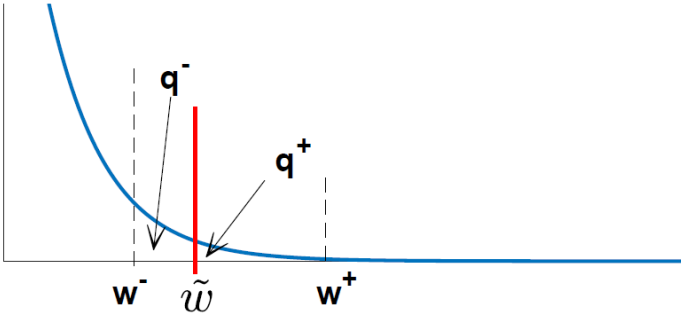

Figure 2.3: Illustration of the procedure for building confidence intervals around the conditional expected weight $\tilde{w} = \langle w_{ij} | a_{ij} = 1 \rangle$.

AIC test will yield the same model ranking as the pure likelihood comparison, since the best performance was awarded to CReM$_B$, followed by CReM$_A$ and finally by the DECM. Since also in terms of complexity the ranking is coincident, the AIC comparison ranks again the CReM$_B$ first, followed by CReM$_A$ and finally by the DECM.
Given the good performance of CReM$_B$ and its low complexity, we recommend this choice and we will soon release a code to make its application fast and straight-forward.

2.3.2 Confidence Intervals-based comparison

In order to introduce the second comparison method, we need to make some observations about the conditional weight distribution $Q(\mathbf{W}|\mathbf{A})$. For all the methods, this distribution factorizes in the product of the individual terms $q_{ij}(w|a_{ij})$. For CReM$_A$, this term reads

$$q_{ij}(w|a_{ij}) = (\beta_i^{out} + \beta_j^{in})^{a_{ij}} e^{-(\beta_i^{out}+\beta_j^{in})w} \tag{2.23}$$

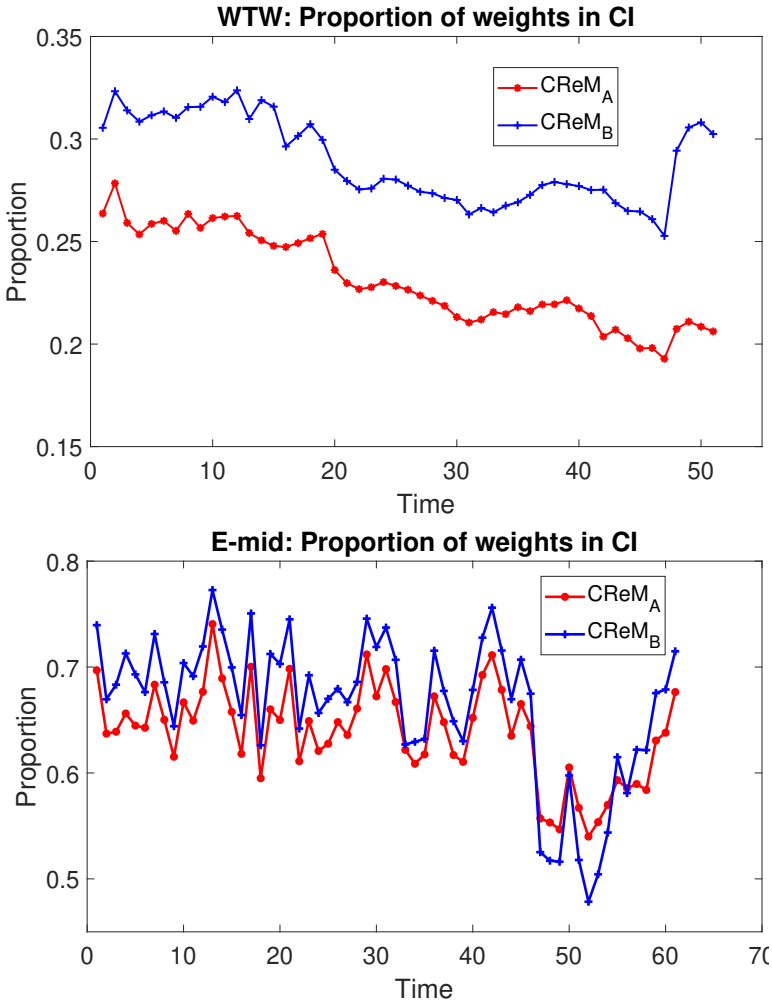

Figure 2.4: Comparison on the proportion of weights falling into the confidence interval relative to the considered method, for the WTW and E-mid datasets respectively, with value $q^+ = q^- = 0.25$. Red line indicates CReM$_A$ and blue line CReM$_B$.

given the compatibility requirement we have $q_{ij}(w|a_{ij} = 0) = \delta_{w,o}$. Moreover we observe that

$$q_{ij}(w|a_{ij} = 1) = (\beta_i^{out} + \beta_j^{in})e^{-(\beta_i^{out}+\beta_j^{in})w}, \tag{2.24}$$

which means that the weight distribution, conditional on the existence of a link, follows an exponential distribution of parameter $\beta_i^{out} + \beta_j^{in}$.
The same observation holds for CReM$_B$, where the exponential parameter is now given by β_{ij}. We observe that both models show the same functional form of the weighted conditional distribution as the DECM (see (1.43)). However the numerical values of the parameters characterizing the distribution are different.[1] Let us denote the parameter of the exponential distribution as λ_{ij} so to keep the reasoning general.
Given the knowledge on the weight distribution we can build confidence intervals around the expected value of the weight. Figure 2.3 gives an illustration of the procedure. The conditional expected value of the weight estimate is indicated as $\tilde{w}$. The blue curve represents the probability density function of the exponential distribution of parameter λ_{ij}. Our goal is to determine the extreme of the confidence interval, w^- and w^+, in such a way that the area under the pdf comprised between the expected weight and w^- is equal to the desired confidence level q^-, and analogously for the upper part of the interval. This is achieved by solving

$$\int_{w^-}^{\langle w_{ij}|a_{ij}=1\rangle} \lambda_{ij} e^{-\lambda_{ij} w_{ij}} dw_{ij} = q^-, \tag{2.25}$$

leading to

$$w^- = -\frac{\ln[e^{-\lambda_{ij}\langle w_{ij}|a_{ij}=1\rangle} + q^-]}{\lambda_{ij}} = -\frac{\ln[e^{-1} + q^-]}{\lambda_{ij}} \tag{2.26}$$

and

$$\int_{w=\langle w_{ij}|a_{ij}=1\rangle}^{w^+} \lambda_{ij} e^{-\lambda_{ij} w_{ij}} dw_{ij} = q^+, \tag{2.27}$$

leading to

$$w^+ = -\frac{\ln[e^{-\lambda_{ij}\langle w_{ij}|a_{ij}=1\rangle} - q^+]}{\lambda_{ij}} = -\frac{\ln[e^{-1} - q^+]}{\lambda_{ij}}. \tag{2.28}$$

In this way, upon deciding on the desired confidence levels q^- and q^+, we are able to define confidence interval for the conditional expected

[1] Although DECM and CReM$_A$ appear to have the same parameters, their values are obtained solving different systems and they will therefore be different in value.

weights. Notice that such a confidence interval is not symmetric, given the peculiar form of the underlying probability distribution (i.e. exponential). Although both the ECM and the DECM can provide an error estimation, its computation is much easier within the novel, continuous framework considered here. Finally, given the confidence intervals for each weight estimate, we can compute the proportion of real weights that fall into the confidence interval relative to their estimate. We can do this for all methods, using the same q^+ and q^- and the different λ_{ij}. The results are shown in Figure 2.4. As for the likelihood-based comparison, model B shows a comparable or better performance than method A, thus confirming our previous considerations. Notice that both the ECM [48] and the DECM provide an error estimation as well; its computation, however, is much easier within the novel, continuous framework considered here.

2.4 Discussion

The method introduced in this chapter aims at filling a methodological gap. Several proposed procedures for network reconstruction, combine different steps to carry out the estimation of topology and of weights, thus introducing biases in the whole procedure. A first source of bias is encountered when a probabilistic recipe for topological reconstruction is forced to produce a *single* outcome instead of considering the entire ensemble of admissible configurations. This choice implies a null likelihood of reproducing the actual network. A second source of bias is encountered when the weights structure is deterministically imposed via a recipe like the RAS one. Again, this recipe ensures a zero likelihood of reproducing the real underlying network: the probability of correctly "guessing" all the link weights is null, assuming continuous weights. Here we reconcile the two aspects, by providing a recipe that clarifies how weights should be determined, once an algorithm for determining the topology of a given network is implemented (be it either probabilistic or deterministic). Notice that the key concept of our approach, i.e. conditional entropy, generalizes traditional approaches which, instead, aim

at jointly determine a given network structure: this is immediately seen by rewriting the probability distribution coming from Shannon entropy maximization as in (2.3).
However, although it is clear how the topology of a network should represent a constrain for the weighted configuration, it is not necessarily true that the weighted configuration has as well an impact of the topology.
The CReM assumes only the first dependency, but allows the network topology to be *unaware* of the weighted configuration.
On a more practical level, the solution of both CReM$_A$ and CReM$_B$ require a lower complexity with respect to the DECM. The latter one, in fact, requires the solution of a system of $4N$ four-wise dependent equation, while CReM$_A$ only involves a system of $2N$ paired equations. CReM$_B$ on the other hand, despite being formally similar in derivation, does not require to solve any system of equations, since the involved parameters are computed via the recipe (2.21). In addition to this, CReM$_B$ shows better or comparable performance with respect the alternative methods, both in terms of likelihood and confidence interval-based comparisons. For this reason, it is the method we recommend choosing.

Chapter 3

Entropy-based approach to missing-links prediction

The work presented in this Chapter is based on the publication [52] by F. Parisi, T. Squartini and G. Caldarelli.

Abstract

Link-prediction is an active research field within network theory, aiming at uncovering missing connections or predicting the emergence of future relationships from the observed network structure. This chapter represents our contribution to the stream of research concerning missing links prediction for binary networks. Here, we propose an entropy-based method to predict a given percentage of missing links, by identifying them with the most probable non-observed ones. The probability coefficients are computed by solving opportunely defined null-models over the accessible network structure. Upon comparing our likelihood based, local method with the most popular algorithms over a set of economic, financial and food networks, we find ours to perform best, as pointed out

by a number of statistical indicators (e.g. the precision, the area under the ROC curve, etc.). Moreover, the entropy-based formalism adopted in the present work allows us to straightforwardly extend the link-prediction exercise to *directed* networks as well, thus overcoming one of the main limitations of current algorithms. The higher accuracy achievable by employing these methods - together with their larger flexibility - makes them strong competitors of available link-prediction algorithms.

3.1 Introduction

Link-prediction is an active research field within network theory, aiming at uncovering missing connections (e.g. in incomplete datasets) or predicting the emergence of future relationships from the observed network structure. Loosely speaking, the missing links prediction problem can be stated by asking the following question: *given a snapshot of a network, can the next most-likely links to be established be predicted?* Such an issue is relevant in many research areas, such as social networks [44, 55, 10, 35], protein networks [8, 60], brain networks [12], etc.

To this aim, several algorithms have been proposed so far. Overall, "recipes" for link-prediction can be classified as belonging to either two main classes, *similarity-based* algorithms or *likelihood-based* algorithms [46, 75]. Both classes of algorithms output a list of scores to be assigned to non-observed links: while the similarity-based ones may employ local [7], quasi-local [12, 34, 61, 57, 1, 76] or global information [38, 47, 75] (e.g. the nodes degree, the degree of common neighbours and the length of paths connecting any two nodes, respectively), the likelihood-based ones [29, 69, 51] are defined by a likelihood function whose maximization provides the probability that any two nodes are connected. This is usually achieved by assuming that some kind of benchmark information is known and by treating it as a constraint to account for. An alternative classification distinguishes between algorithms employing purely structural information (either binary or weighted [46]) and algorithms making

use of some kind of external information as well (e.g. nodes attributes [43]).

This chapter represents our contribution to the stream of research concerning missing links prediction for binary networks. A novel algorithm is proposed, building upon a series of results concerning constrained entropy-maximization [54, 25, 63]. In a nutshell, we advance the hypothesis that the tasks of predicting missing links and reconstructing a given network structure share many similarities worth to be further explored. The method we propose in the present work makes a first step in this direction, by employing entropy-based null-models to approach the link-prediction problem. As a last remark, we notice that while the problem of *missing links prediction* is usually associated to the problem of *spurious links identification*, here we only address the former one.

The remainder of the Chapter is organized as follows. In Section 3.2 an overview of the missing links prediction problem is provided, together with a detailed description of the method we propose here. Section 3.3 contains a synthetic description of the datasets used for testing our methods. In Section 3.4, we compare our method with the most common link-prediction algorithms and we comment on the results in Section 3.5.

3.2 Methods

In order to fix the formalism, let us briefly reformulate the link-prediction problem *ab initio*.

Let us indicate with the symbol $\mathbf{A}$ the adjacency matrix of the observed network and with the symbol E the corresponding set of *observed links*: as a consequence, upon indicating with U the set of all nodes pairs, $U \setminus E$ will be referred to as to the set of *non-existent links*. In order to fully control a given recipe for link-prediction, the link set is usually partitioned into a *training set*, E^T, and a *probe set*, $E^P = E \setminus E^T$. The former is used in the "calibration" phase of a given prediction algorithm, while the latter is used for testing it: links belonging to E^P are, in fact, removed, thus constituting the actual "prediction target". We denote with

$|E^P| \equiv L_{miss}$ the cardinality of the probe set, corresponding to the number of missing links. Naturally, the adjacency matrix is partitioned as well: the portion of it corresponding to the training set will be indicated with the symbol $\mathbf{A}^T$. The union of the missing links set and the non-existent links set $E^N = E^P \cup U \setminus E \equiv U \setminus E^T$ will be referred to as to the set of *non-observed links*.

Link-prediction algorithms output a list of scores to be assigned to non-observed links. Upon indicating with i and j the nodes constituting the extremes of non-observed links, the most traditional recipes are quickly reviewed below. In what follows, we will focus on the algorithms employing either local or quasi-local information.

Link-prediction for undirected networks

- The simplest recipe to define scores is based the number of **common neighbours** (**CN**) of i and j

$$s_{ij}^{CN} = |\Gamma(i) \cap \Gamma(j)|; \tag{3.1}$$

- a slightly more elaborate function of it is represented by the **Jaccard** coefficient (**J**), which discounts the information encoded into the size of the nodes neighbourhoods:

$$s_{ij}^{J} = \frac{|\Gamma(i) \cap \Gamma(j)|}{|\Gamma(i) \cup \Gamma(j)|} = \frac{s_{ij}^{CN}}{k_i + k_j - s_{ij}^{CN}}; \tag{3.2}$$

- algorithms based on the information provided by nodes degrees exist. The simplest example is provided by the one inspired to the **preferential attachment** (**PA**) mechanism, whose generic score reads

$$s_{ij}^{PA} = k_i \cdot k_j; \tag{3.3}$$

- other, instead, are defined by the inverse of some kind of function of the neighbours degree (according to the original **Adamic-Adar** -

AA - prescription or subsequent variations, as the **resource allocation - RA** - one)

$$s_{ij}^{RA} = \sum_{l \in \Gamma(i) \cap \Gamma(j)} \frac{1}{k_l},\ s_{ij}^{AA} = \sum_{l \in \Gamma(i) \cap \Gamma(j)} \frac{1}{\ln k_l}; \tag{3.4}$$

- modifications of the aforementioned indices have been recently proposed, encoding information on the link density of the neighbourhood of each pair of nodes. These indices are the so-called **CAR**-based ones [12] and prescribe to "correct" the scores above by adding a factor $|\gamma(l)|$, counting how many neighbours of node $l \in \Gamma(i) \cap \Gamma(j)$ are also common neighbours of i and j. More explicitly

$$\begin{aligned}
s_{ij}^{CAR} &= s_{ij}^{CN} \cdot \sum_{l \in \Gamma(i) \cap \Gamma(j)} \frac{|\gamma(l)|}{2}, && (3.5)\\
s_{ij}^{CJC} &= \frac{s_{ij}^{CAR}}{|\Gamma(i) \cup \Gamma(j)|}, && (3.6)\\
s_{ij}^{CPA} &= (e_i + s_{ij}^{CAR}) \cdot (e_j + s_{ij}^{CAR}), && (3.7)\\
s_{ij}^{CRA} &= \sum_{l \in \Gamma(i) \cap \Gamma(j)} \frac{\gamma_l}{k_l}, && (3.8)\\
s_{ij}^{CAA} &= \sum_{l \in \Gamma(i) \cap \Gamma(j)} \frac{\gamma_l}{\ln k_l} && (3.9)
\end{aligned}$$

where e_i indicates the external degree of node i, i.e. the number of neighbours of i that *are not* neighbours of j.

Entropy-based approach to link-prediction

The rationale of our method is based upon the concept of *network reconstructability*. In other words, provided that the *accessible* portion $\mathbf{A}^T$ of a network is satisfactorily reproduced by a given amount of topological information, it is reasonable to suppose that the latter allows the *inaccessible* portion to be inferred with reasonable accuracy as well. Invoking the

aforementioned concept allows us to rephrase the link imputation problem within the network reconstruction framework, making it possible to employ the techniques developed there.

From a technical point of view, our algorithm is a local, likelihood-based one. It rests upon the information provided by local, topological quantities, which are enforced as constraints of a maximization procedure defined within the Exponential Random Graph (ERG) framework [54, 63]. In the case of binary, undirected networks, constraints are represented by nodes degrees, i.e. $\vec{k}(\mathbf{A}^T)$ and the ERG framework leads to the maximization of the likelihood function $\mathcal{L} = \ln P(\mathbf{A}^T)$ where

$$P(\mathbf{A}^T) = \prod_{i<j} p_{ij}^{a_{ij}} (1 - p_{ij})^{1-a_{ij}} \tag{3.10}$$

and $p_{ij} = \frac{x_i x_j}{1+x_i x_j}$. The numerical value of the unknown coefficients $\vec{x}$ is obtained upon solving the system of equations

$$k_i(\mathbf{A}^T) = \sum_{j(\neq i)} p_{ij} = \sum_{j(\neq i)} \frac{x_i x_j}{1 + x_i x_j} \; \forall \, i \tag{3.11}$$

(Section 1.3.1 shows the derivation of the condition above in the directed case, the undirected case represents a straightforward simplification). Our algorithm, which is trained on $\mathbf{A}^T$, prescribes to interpret the probability coefficients $\{p_{ij}\}_{ij \in E^N}$ assigned to the non-observed links, as scores to carry out the link-prediction: upon sorting the coefficients $\{p_{ij}\}_{ij \in E^N}$ in decreasing order, the first L_{miss} largest ones are naturally interpreted as pointing out the L_{miss} *most probable missing links* (notice that such a prescription is based on the assumption that the number of missing links is known, although their identity is not: as a consequence, this number is retained). In other words, the reconstructability assumption underlying our method leads us to interpret the non-observed links which have been assigned the largest probability coefficients as the ones that are most likely to appear given the chosen constraints.

Our recipe has a remarkable, equivalent formulation. In fact, the subset $\mathbf{\Sigma}^*$ of L_{miss} links characterized by the largest probability coefficients identifies the subgraph satisfying the relationship

$$\mathbf{\Sigma}^* = \underset{\mathbf{\Sigma}:|E|(\mathbf{\Sigma})=L_{miss}}{\operatorname{argmax}} P(\mathbf{\Sigma}|\mathbf{A}^T) \tag{3.12}$$

with $P(\mathbf{\Sigma}|\mathbf{A}^T) = \prod_{\substack{i<j \\ ij\in E^N}} p_{ij}^{\sigma_{ij}}(1-p_{ij})^{\sigma_{ij}}$. Since the maximum value of such a product is achieved once the L_{miss} largest factors are selected, the generic entry σ_{ij}^* obeys the following rule: $\sigma_{ij}^* = 1$ if ij belongs to the set of L_{miss} most probable missing links and $\sigma_{ij}^* = 0$ otherwise; in other words, $\mathbf{\Sigma}^*$ is the subgraph with largest probability among the ones with precisely L_{miss} links. In the remainder of the chapter, this approach will be named after the null-model employed to calculate the link scores, i.e. UBCM (Undirected Binary Configuration Model) [63].

Link-prediction for directed networks

Remarkably, our algorithm can be generalized to approach the missing links prediction problem in directed networks as well. It is enough to maximize the likelihood $\mathcal{L} = \ln P(\mathbf{A}^T)$ where, now, $P(\mathbf{A}^T) = \prod_{i\neq j} p_{ij}^{a_{ij}}(1-p_{ij})^{1-a_{ij}}$ by solving the system of equations

$$\begin{cases} k_i^{out}(\mathbf{A}^T) &= \sum_{j(\neq i)} p_{ij} = \sum_{j(\neq i)} \frac{x_i y_j}{1+x_i y_j} \;\forall\, i \\ k_i^{in}(\mathbf{A}^T) &= \sum_{j(\neq i)} p_{ji} = \sum_{j(\neq i)} \frac{x_j y_i}{1+x_j y_i} \;\forall\, i \end{cases} \tag{3.13}$$

and consider the coefficients $\{p_{ij}\}_{ij\in E^N}$ as scores to be assigned to the non-observed links (see the Section 1.3.1 for the derivation of the condition above). The proper prediction step is still carried out by applying the recipe defined by eq. 3.12, with the only difference that, now, the product runs over the *directed* pairs of nodes. In the remainder of the chapter, this approach will be named after the null-model employed to calculate the link scores, i.e. DBCM (Directed Binary Configuration Model) [63].

Notice, instead, that no unambiguous ways to generalize traditional scores exist. Here we have adopted the (directed) extensions listed below, with the aim of accounting for link directionality whenever possible:

- when considering directed networks, the concept of **common neighbours** can be replaced by the concepts of "successors" and "prede-

cessors", i.e. the nodes respectively "pointed by" and "pointing to" a given node. Upon indicating the set of "successors" of i with Γ_S and the set of "predecessors" of j with Γ_P, the CN index can be generalized as follows

$$s_{ij}^{CN} = |\Gamma_S(i) \cap \Gamma_P(j)|; \tag{3.14}$$

- building upon the directed version of the CN index, the **Jaccard** index reads

$$s_{ij}^{J} = \frac{|\Gamma_S(i) \cap \Gamma_P(j)|}{|\Gamma_S(i) \cup \Gamma_P(j)|} = \frac{s_{ij}^{CN}}{k_i^{out} + k_j^{in} - s_{ij}^{CN}}; \tag{3.15}$$

- the **RA** and **AA** indices can be straightforwardly generalized as follows:

$$s_{ij}^{RA} = \sum_{l \in \Gamma(i) \cap \Gamma(j)} \frac{1}{k_l^{tot}},\ s_{ij}^{AA} = \sum_{l \in \Gamma(i) \cap \Gamma(j)} \frac{1}{\ln k_l^{tot}} \tag{3.16}$$

  with $k_i^{tot} = k_i^{out} + k_i^{in}$;

- the **PA** score admits two different generalizations: one employing the total degree of nodes

$$s_{ij}^{PA'_I} = k_i^{tot} \cdot k_j^{tot} \tag{3.17}$$

  and the other employing the nodes out- and in-degree

$$s_{ij}^{PA'_{II}} = k_i^{out} \cdot k_j^{in}; \tag{3.18}$$

- while the CAR-based indices are not straightforwardly generalizable to the directed case, other scores exist aiming at extending the concept of "closed triad" to account for link directionality [59], the **triadic closure** index (**TC**) is defined as:

$$s_{ij}^{TC} = \sum_{l \in \Gamma(i) \cap \Gamma(j)} w_{i,j,l} \cdot w(l); \tag{3.19}$$

here, the "triad weight" $w_{i,j,l} = \frac{\#T_{i\to j,l}+\#T_{i\leftrightarrow j,l}}{\#T_{i,j,l}}$ is defined by the (global) number $\#T_{i,j,l}$ of observed, *open* triads of the particular kind $T_{i,j,l}$, the (global) number $\#T_{i\to j,l}$ of observed, *closed* triads via a directed link from i to j and the (global) number $\#T_{i\leftrightarrow j,l}$ of observed, *closed* triads via a reciprocal link between i and j; $w(l)$ is, instead, a node-specific weight that can be set either to $w(l) = \frac{1}{k_l}$ or to 1. In order to avoid misinterpretations, we set the weight to 1.

Testing link-prediction

Once a link-prediction algorithm has been defined, a number of statistical indices exist to test its effectiveness. In what follows we will briefly review the ones we have employed in the present work to compare the aforementioned algorithms. The first index we have considered is the *true positive rate* (also known with the name of *precision*), defined as

$$\text{TPR} = \frac{L_r}{L_{miss}} \tag{3.20}$$

and quantifying the percentage of missing links that are correctly recovered (i.e. the number L_r of rightly identified missing links within the list of the first L_{miss} links with the largest score). A similar-in-spirit index is the *accuracy*

$$\text{ACC} = \frac{L_r + L_{ne}}{|E^N|}, \tag{3.21}$$

quantifying the percentage of correctly classified links (i.e. both the missing ones and the non-existent ones) with respect to the total number of non-observed links. The third index we consider is the traditional *area under the ROC curve*, or AUC, proxied by the number

$$\text{AUC} = \frac{n' + n''/2}{n}; \tag{3.22}$$

n' counts the number of times a missing-link is assigned a higher probability than to a non-existent one, while n'' accounts for the number of times they are assigned an equal probability. The denominator n coincides with the total number of comparisons (i.e. the number of missing

links times the number of non-existent links). This index is intended to quantify the probability that any missing-link is assigned a score that is larger than the score assigned to any non-observed link. If all scores were i.i.d. the AUC value should be distributed around an expected value of $1/2$: therefore, the extent to which the AUC value exceeds 0.5 provides an indication of how much better the algorithm performs than pure chance.

The set of missing links is usually randomly removed: we have followed such a procedure, by 1) randomly removing the 10% of links 10 times, 2) quantifying the performance of the algorithms above, by computing the three aforementioned indices over each sample, 3) averaging these values over the sample set (the sample standard deviation is used to proxy the estimation error.)

3.3 Data

Our approach to link-prediction has been tested on a number of economic and financial datasets (see table 3.1) and on several food-webs (see table 3.2).

As a first dataset, we have considered the World Trade Web (WTW) across a period of 51 years, i.e. from 1950 to 2000. The dataset in [26] collects yearly, bilateral, aggregated data on exports and imports (the generic entry $m_{ij}^{agg}(y)$ is the sum of the single commodity-specific trade exchanges between i and j during the year y). The binary, directed representation of the WTW we have considered here has been obtained by linking any two nodes whenever the corresponding element $m_{ij}^{agg}(y)$ is strictly positive, i.e. $a_{ij}(y) = \Theta[m_{ij}^{agg}(y)]$.

As a second dataset, we have considered the Dutch Interbank Network (DIN) across a period of 11 years, i.e. from 1998 to 2008 [41]. Such a dataset collects quarterly data on exposures between Dutch banks, larger than 1.5 million euros and with maturity shorter than one year.

As a third dataset, we have considered the e-MID (i.e. the electronic Market for Interbank Deposits) network in a series of 61 temporal snapshots, corresponding to the maintenance periods (and ranging from 2005

World Trade Web			
BUN	N	L	$\langle k \rangle$
	145 [85,187]	5901 [1678,10254]	76.3 [39.5,109.7]
BDN		L	$\langle k \rangle$
		10604 [2871,20107]	68.2 [33.8,107.5]
e-MID			
BUN	N	L	$\langle k \rangle$
	152 [107,170]	1755 [742,2333]	22.5 [13.4,28.6]
BDN		L	$\langle k \rangle$
		1827 [754,2477]	11.7 [6.8,14.9]
Dutch Interbank Network			
BUN	N	L	$\langle k \rangle$
	99 [92,104]	675 [444,1025]	13.6 [8.9,20.1]
BDN		L	$\langle k \rangle$
		773 [512,1207]	7.8 [5.2,11.8]

Table 3.1: Network statistics for both the undirected (BUN) and directed (BDN) version of the World Trade Web, e-MID and the Dutch Interbank Network: for each quantity (number of nodes N, number of links L and average degree $\langle k \rangle$) the mean value across the temporal snapshots and the range are reported.

Food-webs	N	L	$\langle k \rangle$
Chesapeake Bay	39	177	4.54
Lower Chesapeake Bay	37	178	4.81
Middle Chesapeake Bay	37	209	5.65
Upper Chesapeake Bay	37	215	5.81
Everglades Marshes	69	916	13.28
Florida Bay	128	2106	16.45
Grassland	88	137	1.56
Little Rock Lake	183	2494	13.63
Maspalomas Lagoon	24	82	3.42
Michigan Lake	39	221	5.67
Mondego Estuary	46	400	8.70
Narragansett Bay	35	220	6.29
Rhode River Watershed	19	53	2.79
Silwood Park	154	370	2.40
St Marks River	54	356	6.59
St Marks Seagrass	49	226	4.61
St Martin Island	45	224	4.98
Ythan Estuary	135	601	4.45

Table 3.2: Network statistics (number of nodes N, number of links L and average degree $\langle k \rangle$ for the directed version of 18 different food-webs [63].

to 2010). In this case, links represent granted loans [33]. As for the WTW, the binary, directed representations of both the DIN and e-MID have been obtained by linking any two nodes whenever a positive weight is observed between them.

The three real-world systems above are defined by directed connections. In order to evaluate the performance of the link-prediction algorithms considered in the present work on undirected networks, we have properly symmetrized the adjacency matrices of these systems, according to the prescription $a_{ij}^{sym} = a_{ij} + a_{ji} - a_{ij}a_{ji}$.

Food-webs, instead, are considered in their binary, directed version only: if species i preys on species j, a directed link is drawn from j to i.

3.4 Results

The performance of our link-prediction algorithm is shown in Figures 3.1 to 3.5 for the WTW, the DIN and e-MID datasets respectively while te results for the food-webs are reported in Figure 3.6. Figure 3.1 has been enlarge to facilitate the understanding, all other figures are organized in an analogous way.

As a general comment, our method performs better than the other algorithms, with respect to all considered indices. The success of the method is particularly evident when considering the AUC index, proxying the probability of (correctly) assigning a larger score to a missing-link than to a non-existent link.

We argue the success of our algorithm to rest upon a core result that has been verified in a number of previous works [63, 62, 13]: the (purely) topological structure of the networks considered here can be reconstructed, to a large degree of accuracy, by enforcing the information encoded into the degree sequences alone; very likely, thus, the same amount of information also defines an accurate recipe to spot potential missing links. Otherwise stated, the level of "complexity" of the considered networks seems to be largely encoded into the degree sequences, thus requiring (just) their enforcement to be fully accounted for.

The founding principle of our approach is, thus, radically different from the one inspiring other link-prediction algorithms: we aim at finding the (most likely) generative process for the network at hand, while other methods define increasingly detailed procedures with little control on the "quality" of the included information. This becomes evident when considering that other algorithms (i.e. the CN, J, RA, AA and the CAR-based ones) employ a larger amount of information than the UBCM- and DBCM-based ones: while the latter take as input just the nodes degrees, the former exploit the information provided by the whole set of common neighbours. This may indicate that the information encoded into the neighbourhood of any two nodes - supposedly providing more information than the one encoded into the degrees alone - is, actually, a mere consequence of lower-order statistics (i.e. the degrees

themselves).

This also sheds light on the reason why our algorithm is less sensitive than others to the original value of link density: provided that our entropy-based recipes successfully individuate the process generating the networks at hand, the number of observed links is automatically accounted for.

Our comparison also points out that one of the factors determining the goodness of a given link-prediction algorithm concerns *how* the available information is used. An illustrative example is provided by the performance of the PA algorithm defined by eq. 3.17, requiring the same basic knowledge of our entropy-based recipes, i.e. the degree sequences of nodes. As clear upon inspecting the e-MID directed case, the assumption that any two nodes establish a connection with a probability that is proportional to their *total* degrees fails to capture the process shaping the network structure; entropy-maximization, on the other hand, makes a better use of the available information, by retaining the information on link directionality (that indeed plays a role, completely ignored by the aforementioned PA prescription).

Interestingly, the DBCM recipe described in the Section 3.2 induces the "correct", directed generalization of the PA algorithm, defined by eq. 3.18 and outperforming the one defined by eq. 3.17. For sparse networks, in fact, the DBCM probability coefficients can be approximated as follows

$$p_{ij}^{\text{DBCM}} \simeq \frac{k_i^{out} \cdot k_j^{in}}{L} \tag{3.23}$$

a simplified prescription that performs very similarly to the entropy-based algorithm on the DIN and e-MID; on dense networks - as the WTW - the DBCM performs much better, instead. The UBCM, on the other hand, reduces to $p_{ij}^{\text{UBCM}} \propto k_i \cdot k_j$, i.e. the undirected PA prescription defined by eq. 3.3.

Finally, let us comment on the performance of the TC index. The algorithm employing it has been designed to provide a solution to the problem of forecasting new connections among social networks users. By definition, it only predicts new links among *disconnected* nodes, disre-

garding all nodes pairs connected by, e.g. a non-reciprocated link. This explains the poor performance of the algorithm in our context, despite it performs satisfactorily to solve the specific task it was designed for [59].

3.5 Discussion

Whenever judging the performance of a given link-prediction algorithm, one should consider both the amount of information it requires and the way in which this is employed to carry out the prediction step. While the usual link-prediction algorithms assume the existence of some node-specific tendency at a microscopic level (e.g. social agents tend to close triads), ours focuses on the most likely process that may have generated the considered network. The guessed process is, first, trained on the visible portion of the network and, then, employed to infer the (supposedly) unknown portion of the network: the "homogeneity" assumption underlying the whole procedure leads us to expect that a model satisfactorily reproducing the accessible part of a system is also effective in spotting potential missing links.

One of the most effective recipes to tune generative processes is the one based on the entropy-maximization: beside guaranteeing that the available information is encoded in the least-biased way, the ERG framework is also very flexible, being applicable to both undirected and directed networks; other algorithms, on the contrary, rest upon concepts unambiguously defined only for undirected networks (an example is provided by the whole family of CAR indicators, whose core concept - i.e. the "local community links" factor $|\gamma(l)|$ - does not admit a straightforward generalization).

Although every newly-proposed algorithm fosters the idea to be applicable to different kinds of systems, the effectiveness of a given (null) model depends on the particular system at hand: while economic, financial and food networks seem to be largely explained by the degree sequences, other systems may require a different (or additional) kind of information.

The results obtained so far on undirected, as well as directed, binary

networks push us to look for further extensions of the proposed link-prediction technique. Interesting perspectives are represented by bipartite and weighted networks, for which the link- and weight-imputation topics are still little explored. This will be the topic of the next chapter.

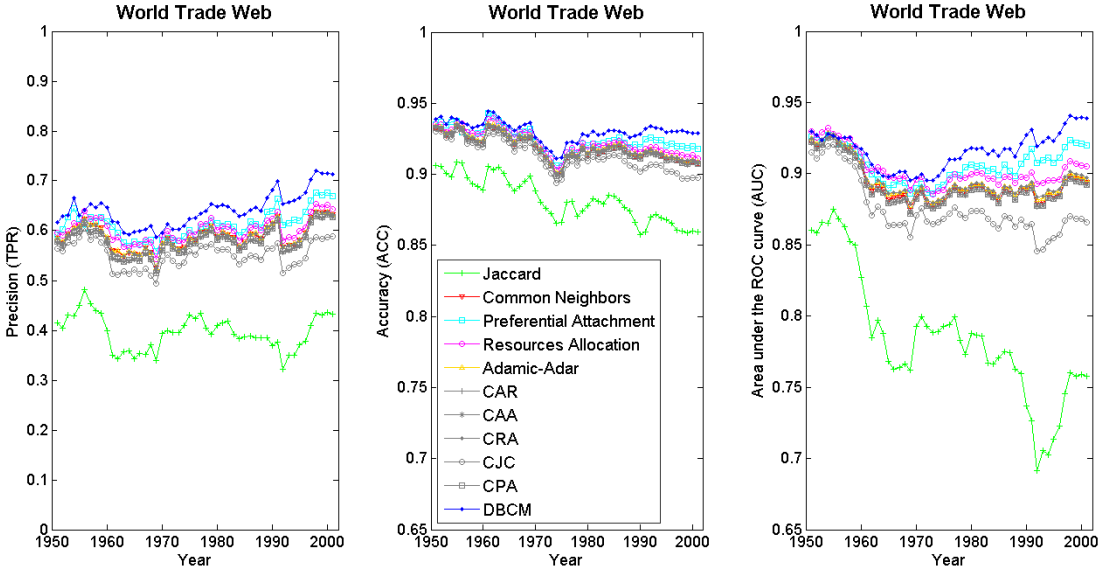

Figure 3.1: Comparison of link-prediction algorithms when applied to the undirected version of the WTW. Left panel: evolution of precision; middle panel: evolution of accuracy; right panel: evolution of AUC. Our entropy-based approach (dark blue line) to link-prediction performs better than other algorithms, across all temporal snapshots.

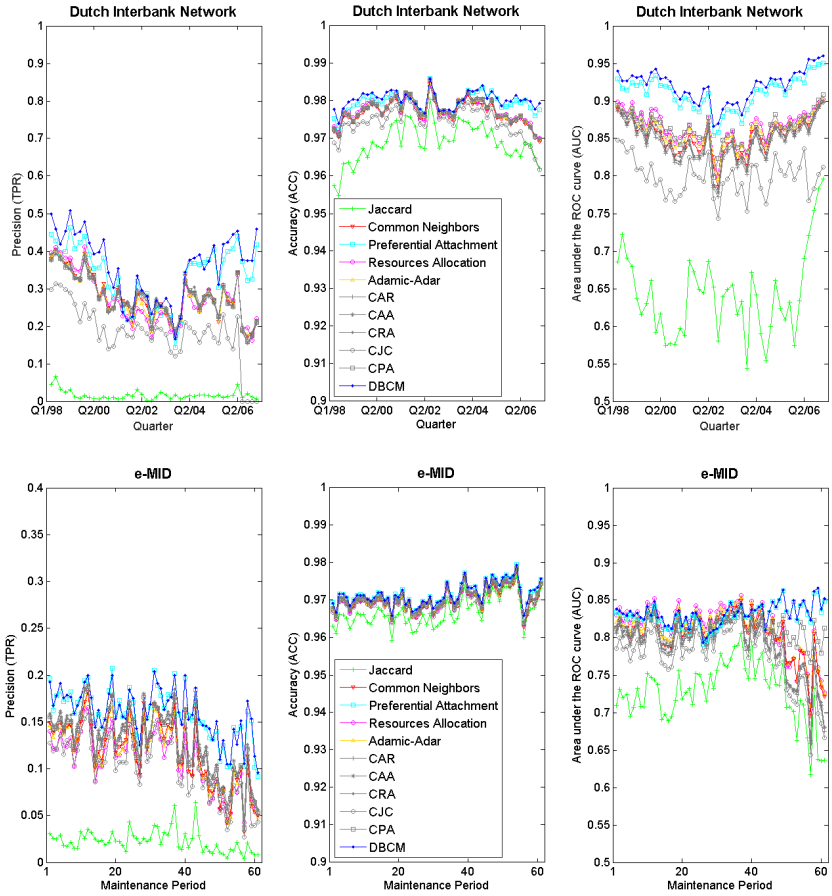

Figure 3.2: Comparison of link-prediction algorithms when applied to the undirected version of the DIN (top) and E-mid (bottom). The panels are organized as in Figure 3.1. Our entropy-based approach (dark blue line) to link-prediction performs better than other algorithms, across the vast majority of temporal snapshots.

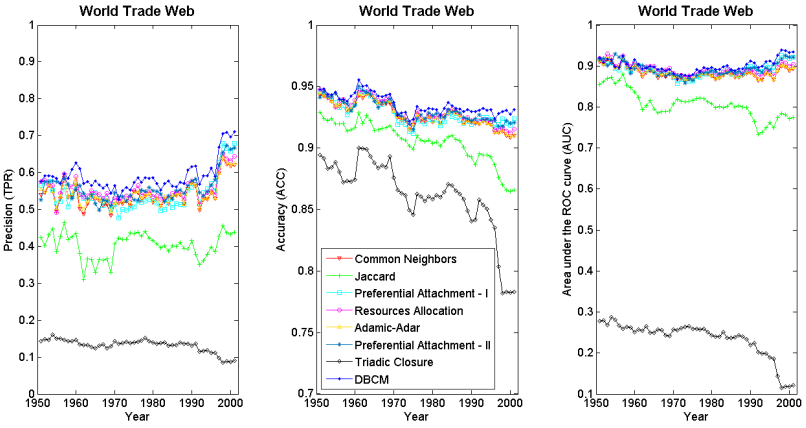

Figure 3.3: Comparison of link-prediction algorithms when applied to the directed version of the WTW. The panels are organized as in Figure 3.1. Our entropy-based approach (dark blue line) to link-prediction performs better than other algorithms, across the vast majority of temporal snapshots.

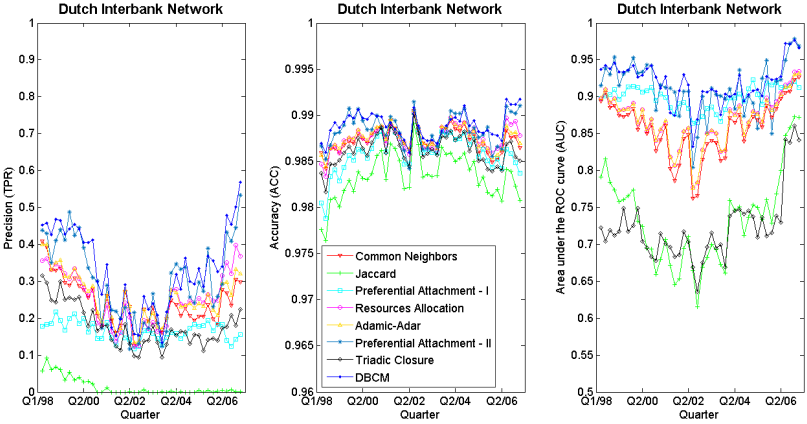

Figure 3.4: Comparison of link-prediction algorithms when applied to the directed version of e-MID. The panels are organized as in Figure 3.1. Our entropy-based approach (dark blue line)to link-prediction performs better than other algorithms, across the vast majority of temporal snapshots.

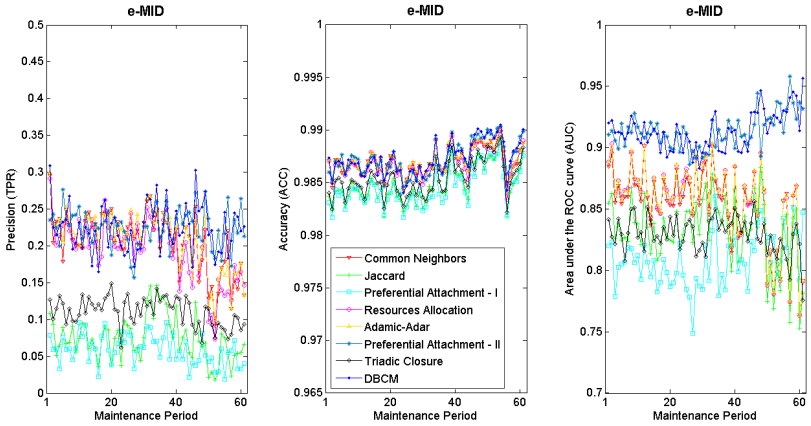

Figure 3.5: Comparison of link-prediction algorithms when applied to the directed version of the e-MID. The panels are organized as in Figure 3.1. Our entropy-based approach (dark blue line) to link-prediction performs better than other algorithms, across the vast majority of temporal snapshots.

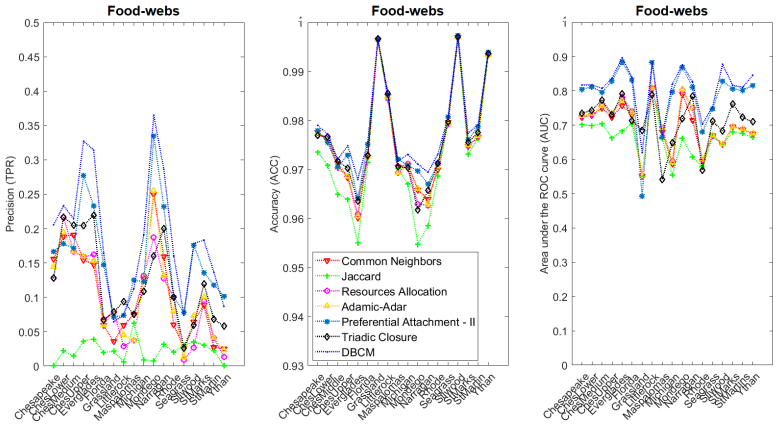

Figure 3.6: Comparison of link-prediction algorithms when applied to the (directed version) of the food-webs. Our entropy-based approach to link-prediction performs better than the other algorithms. Here, we have retained only the recipe $s_{ij}^{PA''} \propto k_i^{out} \cdot k_j^{in}$.

Chapter 4

Two extensions to the missing-link prediction framework

The part of this chapter on bipartite link prediction is based on [6]
by M. Baltakiene, K. Baltakys, D. Cardamone, F. Parisi, T. Radicioni, M. Torricelli, J. A. van Lidth de Jeude, and F. Saracco
Result of a project carried out during the workshop `Complexity72h`
Lucca 7-11 May 2018
The part concerning link and weights prediction on weighted networks is the result of joint work with T. Squartini and has not been published yet.

Abstract

In this chapter we aim at broadening the spectrum of applicability of the entropy-based methods for link prediction by moving in two different directions: we will consider weighted and bipartite networks. In the first case the problem of link prediction is two-fold: not only we need to identify the "identity" of the missing link, but we also need to decide which

weight to attribute to the freshly imputed links. The vast majority of methods for link prediction in weighted networks, however, does not provide a procedure for weight estimation. Here we propose a new approach based on the idea that the *reconstructability* of a network guarantees that certain network characteristics, as the most probable missing link or the value of the link weights, can successfully be inferred. We discuss different possible choices concerning the null model used: we will show that entropy-based methods have better performances with respect to the alternatives and they also provide, unlike the others, weight estimates and the corresponding confidence intervals. The second extension we deal with in this chapter is the one to bipartite networks. Here we apply as a null model the Bipartite Configuration Model (BiCM) that represents the natural extension to the bipartite case of the DBCM model used in the case of binary monopartite networks. We will show that our method has a very good performance. Remarkably, in cases where it is not the "race winner", it achieves a comparable performance with respect to alternative algorithms, but employing a significantly smaller amount of information.

4.1 Introduction

In the previous chapter we introduced the problem of link prediction, its importance and some of the current techniques to tackle this issue. However, we focused only on binary monopartite networks. This is the main limitation of the methods currently present in the literature on the topic: they are devised to be applied to the simplest network structures and are not readily applicable to other scenarios. In this chapter we aim at broadening the spectrum of applicability of the entropy-based methods for link prediction by moving in two different directions: we will consider weighted and bipartite networks.
In the first case the problem of link prediction is two-fold: not only

we need to identify the missing link, but we also need to decide which weight to attribute to the freshly imputed links. The vast majority of methods for link prediction in weighted networks, however, does not provide a procedure for weight estimation, thus limiting themselves to the task of inferring the network topology from weights. Amid these methods, one uses standard similarity measure as predictors for a supervised machine learning algorithm [56]. This methods is focused on social networks and aims at exploring whether adding weight information improves the link prediction performance. The results are inconclusive since it appears that the link prediction performance is improved by the weighted information only on certain datasets. In [73] the authors extend their previous work, which used path entropy as a score for link prediction, to accomodate weighted information. The authors of [77] use a mutual information approach for link prediction and explore the relative importance of weak and strong ties for the inference of missing links. In [72] a bayesian framework is applied by proposing a weighted version of the Local Naive Bayes algorithm. The authors focus their comparison on social networks, in particular collaboration networks.

Among the methods that allow for proper weight inference , in [75] the estimation is based on the consideration that the link weights are proportional to the similarity measures used for the binary imputation. However they estimate this proportionality parameter *a posteriori* using the - in principle - unobservable part of the data, thus making the method inapplicable to real missing links problems.

In another work [49], the authors devise an interesting, and fairly complex, method for weight estimation, which is uses in combination with some standard link prediction measures. The method is tailored specifically for social networks but can be applied to other domains. The main drawback of this method is that the binary part of the prediction, that is the identification of the missing links, is carried out using the weights themselves as a similarity measure. This choice implicitly assumes that the existence and intensity of a link derive from the same process, in opposition to other stances in the literature [24]. For more details see the discussion on the interplay between binary topology and weighted

structure in Chapter 2. In this work we will focus on local (node specific) similarity-based algorithms.

We propose a new approach based on methods developed for the task of network reconstruction. As in the binary case, the key idea is that the *reconstructability* of a network guarantees that certain network characteristics, as the most probable missing link or the value of the link weights, can successfully be inferred. However, in the weighted case we discuss several possible choices concerning the null model used to reconstruct the network: one, the Directed Enhanced Configuration Model (DECM) [15], that assumes that the binary and weighted structure of a network are jointly determined in the network formation process. Moreover, we apply the two specifications of the CReM method, CReM$_A$ and CReM$_B$, presented in Chapter 2. We will show that all three entropy-based methods have better performances with respect to the alternatives and they also provide, unlike the others, a method to obtain estimates for the link weights and their confidence interval. We will also show that CReM$_B$ is characterize by a better or comparable performance with respect to CReM$_A$ and DECM, and a much lower complexity. We will therefore recommend the choice of CReM$_B$ for the task of link and weights prediction.

The second extension we deal with in this chapter is the one to bipartite networks. Many real-world system can be represented as bipartite networks [28, 40], such as collaboration and co-authorships networks [50], recommendation networks [45, 9], financial networks of banks and assets [27], biological mutualistic networks [20, 68, 4] and trade networks [32, 18]. Standard approaches for their analysis quite often rely on the projection on one of the layers, but the information contained in the original bipartite network can provide important insights for the comprehension of the phenomena under analysis [58]. However, only few of the methods proposed in the literature have been applied in the case of bipartite networks[74, 19, 23]. Among the algorithms which admit bipartite configurations, there are several classes of techiques, such as global and

kernel-based methods[39], extensions of results in monopartite networks to bipartite [19] and projections on the monopartite [23, 74].

Here we apply as a null model the Bipartite Configuration Model (BiCM) that represents the natural extension to the bipartite case of the DBCM model used in the case of binary monopartite networks. We will show that our method has a very good performance. Remarkably, in where cases it is not the "race winner", it achieves a comparable performance with respect to alternative algorithms, employing a significantly smaller amount of information.

This chapter is organized as follows: Section 4.2 introduces the methods used for weighted link prediction, starting from the methods from the literature and then presenting and discussing the three options for the maximum entropy methods. Section 4.3 has an analogous outline for the methods concerning bipartite link prediction. Section 4.3 describes the bipartite datasets, while the datasets for weighted prediction coincide with the ones used in the previous chapter. Section 4.4 presents the results of the weighted prediction first, and of the bipartite one successively. Finally, Section 4.5 presents the final remarks.

4.2 Methods for Weighted Networks

We will use the same notation as in the previous chapter concerning the link prediction problem definition. As far as comparison methods are concerned, we compare with some methods taken from the literature [77], extending to the weighted case some standard metrics. As in the binary case, we focus on the algorithms employing either local or quasi-local information.

Similarity measures for weighted directed networks

- **Weighted Common neighbors (WCN)**: The common neighbours index can be extended to the weighted case (WCN) by computing,

between nodes i and j, the quantity

$$s_{ij}^{WCN} = \sum_{z \in S(i) \cap P(j)} (w_{iz} + w_{zj}), \tag{4.1}$$

where $S(i)$ is the sut of successors of node i, that is all the nodes k such that there is a link from i to k, while $P(j)$ is the set of predecessors of node j, that is all nodes h such that there is a link from h to j. The intersection of these sets gives all the nodes that are the middle node in the paths of length two from i to j.

- **Weighted Jaccard (WJ)**: the weighted Jaccard coefficient (WJ), which discounts the information encoded into the size of the nodes neighbourhoods, reads:

$$s_{ij}^{WJ} = \sum_{z \in S(i) \cap P(j)} \frac{w_{iz} + w_{zj}}{s_i^{out} + s_j^{in}}; \tag{4.2}$$

- **Weighted Preferential Attachment (WPA)**: the preferential attachment rule, that prescribes the probability of a link to be quantified by the product of the link degrees, can be modified by considering the link strengths

$$s_{ij}^{WPA} = s_i^{out} * s_j^{in}; \tag{4.3}$$

- **Weighted Adamic-Adar (WAA)**: another common neighbours-based measures, defined by the inverse of a function of the neighbours degree, the original Adamic-Adar , can be extended to account for the node strengths

$$s_{ij}^{WAA} = \sum_{z \in S(i) \cap P(j)} \frac{w_{iz} + w_{zj}}{\log(1 + s_z^{tot})}; \tag{4.4}$$

- **Weighted Resource Allocation (WRA)**: Similar in spirit to the Adamic-Adar index, the Resource Allocation uses a different averaging tech-

nique.

$$s_{ij}^{WRA} = \sum_{z \in S(i) \cap P(j)} \frac{w_{iz} + w_{zj}}{s_z^{tot}}, \tag{4.5}$$

Entropy-based approach to weighted link-prediction

As for the binary case, the idea at the base of our link prediction method is that a model that is able to reconstruct the properties of a network given aggregate information can also be applied to the identification of the most probable missing links. This is achieved by computing the linkage probabilities through the desired reconstruction method and then drawing the unobserved link with highest probability. In the case of weighted networks, though, the identification of the missing link does not exhaust the problem: it is also necessary to assign a weight to the imputed links. While in the previous chapter we had a single choice for the entropy based reconstruction method, here we propose different alternatives. They encode different assumptions on the network formation process and on the weight structure. Specifically, we will use

DECM The Directed Enhanced Configuration Model, continuous case;

CReM$_\mathbf{A}$ The Conditional Reconstruction Method, specification A;

CReM$_\mathbf{B}$ The Conditional Reconstruction Method, specification B.

These three methods have been introduced in the previous chapters, so we will briefly recall their main specifications.

Directed Enhanced Configuration Model The DECM is based on the assumptions that the topology and weighted structure of a network are jointly determined in the network formation process. This method chooses as constraints the in- and out- degree and strength sequences, that will be preserved, on average, over the ensemble of weighted configurations. The quantities $\vec{x}^{out}$ and $\vec{x}^{in}$ are (a transformation of) the Lagrange multipliers associated to the binary constraints (i.e. degree sequences), while $\vec{\beta}^{out}$ and $\vec{\beta}^{in}$ are the Lagrange multipliers associated with the weighted

constraints (i.e. strength sequences). The DECM model is characterized by the linkage probability (i.e. the probability of a link existing from node i to node j) given by

$$p_{ij}^{DECM} = \left[1 + (\beta_i^{out} + \beta_j^{in})/(x_i^{out} x_j^{in})\right]^{-1}, \tag{4.6}$$

while the probability of observing a link of weigh w, given the existence of a link from i to j reads

$$q_{ij}^{DECM}(w > 0) = p_{ij}^{DECM}(\beta_i^{out} + \beta_j^{in}) e^{-(\beta_i^{out} + \beta_j^{in})w}. \tag{4.7}$$

As a consequence, the expected weight is

$$\langle w_{ij} \rangle^{DECM} = \frac{p_{ij}^{DECM}}{\beta_i^{out} + \beta_j^{in}} \;\Rightarrow\; \langle w_{ij} | a_{ij} = 1 \rangle^{DECM} = \frac{1}{\beta_i^{out} + \beta_j^{in}}, \tag{4.8}$$

where the last expression is the conditional expected weight, that is the value of the expected weight given that a link is present.

Conditional Reconstruction Method The CReM models rely on a different assumption, regarding the network formation process, with respect to the DECM. In this case, in fact, we consider the network topology to be generated first, as a realization of a random variable, and then the weighted configuration is determined, compatibly with the given topology. As explained in Chapter 2, the CReM takes as input a distribution over the space of binary configurations and weighted constraints. Many choices are available for the computation of this binary distribution. Given the good results obtained in the previous chapter, here we choose to use the linkage probabilities given by the Directed Binary Configuration Model (DBCM). Therefore for both CReM$_A$ and CReM$_B$ we have

$$p_{ij}^{DBCM} = \frac{x_i^{out} x_j^{in}}{1 + x_i^{out} x_j^{in}}, \tag{4.9}$$

where, as before, $\vec{x}^{out}$ and $\vec{x}^{in}$ are (a transformation of) the Lagrange multipliers associated to the binary constraints (i.e. degree sequences).
Let us specify the notation: when we use the same letter to denote the

parameters of different models, we do so to stress the fact that they play the same role, that is for instance they are the Lagrange multipliers associated to the same constraints. However the same parameters will have different values in different models, since their value will be obtained solving different systems. [1]

As far as the weight estimation is concerned the two model specification provide different recipes.

A. The Conditional Reconstruction Method, specification A (CReM$_A$) uses the observed strength sequences as constraints, keeping the ensemble average of such quantities equal to the observed value. The weight distribution for this model takes the form

$$q_{ij}^{CReM_A}(w>0) = p_{ij}^{DBCM}(\beta_i^{out}+\beta_j^{in})e^{-(\beta_i^{out}+\beta_j^{in})w}. \tag{4.10}$$

This differs from (4.7) in two ways: first of all the binary probability is different, since CReM$_A$ uses the linkage probabilities (4.9). Moreover, as previously observed, the values of the parameters $\vec{\beta}$ will be different for DECM and CReM$_A$. From (4.10) we have

$$\langle w_{ij}\rangle^{CReM_A} = \frac{p_{ij}^{DBCM}}{\beta_i^{out}+\beta_j^{in}} \;\Rightarrow\; \langle w_{ij}|a_{ij}=1\rangle^{CReM_A} = \frac{1}{\beta_i^{out}+\beta_j^{in}}, \tag{4.11}$$

B. The Conditional Reconstruction Method, specification A (CReM$_B$) imposes that the ensemble average of the link weights is equal to the quantity that has been empirically shown to be a good proxy for the link weights, that is the product of the node strengths over the total weight of the network: $w_{ij} \sim \frac{s_i^{out}s_j^{in}}{W_{tot}}$. The functional form of the weight distribution is

$$q_{ij}^{CReM_B}(w>0) = p_{ij}^{DBCM}(\beta_{ij})e^{-\beta_{ij}w}. \tag{4.12}$$

[1] To be fully precise we should use a notation of the kind $\vec{x}^{DECM}$ and $\vec{x}^{DBCM}$. However this would make the formulae very difficult to read. Therefore we will simply use the same parameter notation since it will be clear to which model they refer.

Imposing that the expected weights correspond to the desired quantities yields

$$\langle w_{ij}\rangle^{CReM_B} = \frac{s_i^{out}s_j^{in}}{W_{tot}} \Rightarrow \langle w_{ij}|a_{ij}=1\rangle^{CReM_B} = \frac{s_i^{out}s_j^{in}}{W_{tot}p_{ij}^{DBCM}}. \tag{4.13}$$

Although this method is formally derived in an analogous way with respect to CReM$_A$, it does not require the solution of a system of equation for determining the value of the matrix of parameters $\boldsymbol{\beta}$, since (4.12) and (4.13) imply

$$\beta_{ij} = \frac{W_{tot}p_{ij}^{DBCM}}{s_i^{out}s_j^{in}}. \tag{4.14}$$

For all three methods the link prediction procedure follows the same steps

1. We solve the model and obtain p_{ij} and $\langle w_{ij}|a_{ij}=1\rangle$.
2. We rank the p_{ij} probabilities in descending order.
3. We draw the L_{miss} *unobserved* links with highest probability.
4. We assign to the imputed link the conditional expected weights $\langle w_{ij}|a_{ij}=1\rangle$.

Let us observe that, in principle, we could have decide to use as score for the link prediction the q_{ij} probability evaluated in $\langle w_{ij}|a_{ij}=1\rangle$. This would correspond to compute the probability of observing a weight equal to the estimated one. Testing this alternative approach showed a diminished performance. This indicates that to predict the topology alone the linkage probability should be preferred over the overall one.

Testing link and weights predictions

To test the effectiveness of our link prediction methods against the alternatives, we need to consider the performances on two different tasks: the binary link prediction and the weight estimation. With respect to the first

task, we will use the same indicators introduced in the previous chapter: accuracy, precision and area under the ROC curve (AUC).
As far as the weight estimate is concerned, first we remark once again that the comparison methods do not allow to estimate the link weights, therefore we will compare only the three entropy-based models. We will compare only the imputed part of the matrix with the real counterpart, since the rest of the matrices will coincide. We denote with w_{ij}^{obs} the real observed weights and with w_{ij}^{est} the model estimates, relative to the imputed part of the matrix. Since some measures are designed to take as input vectors and not matrices, we will vectorialize the weights matrices, denoting them as $\vec{w}^{obs}$ and $\vec{w}^{est}$. To quantify the models relative performances we will use two quantities:

Cos_sim The first indicator is simply the cosine similarity between the vectors of real weights and estimated ones. Since we want a measure that is disentangled from the correctness of the binary prediction, we only consider the couples of weights where both the real ad imputed weights are non-zero.

$$\text{Cos\_sim} = \frac{\vec{w}^{obs} \cdot \vec{w}^{est}}{\|\vec{w}^{obs}\|_2 \cdot \|\vec{w}^{est}\|_2} \tag{4.15}$$

Prop_CI The second indicator requires a bit more explanation. As we can see from the expressions (4.7), (4.10) and (4.12), for all three methods the distribution of weights conditional on the existence of a link follows an exponential distribution, whose parameters are model specific. This observation allows us to compute confidence intervals for the weight estimates. (See Section 2.3.2 in Chapter 2 for details.) Therefore we can use as a second indicator the proportion of real weights that fall into the confidence interval relative to their estimate.

$$\text{Prop\_CI} = \frac{|w_{ij}^{obs} \in \text{CI}\{w_{ij}^{est}\}|}{L\_miss}, \tag{4.16}$$

where N is the number of nodes.

The evaluation procedure was carried out in the following way:

1. Starting from a fully observable weighted adjacency matrix, $\mathbf{W}$, we remove 10% of its links at random, creating the incomplete matrix $\widehat{\mathbf{W}}$.

2. For any given method m we run the link prediction algorithm that returns a matrix $\mathbf{R}_m$, with the same number of links than $\mathbf{W}$.

3. We compute all the evaluation measures on $\mathbf{R}_m$.

4. We repeat steps 1. to 3. for 10 times and we compute the average of the evaluation measures.

4.3 Methods for Binary Bipartite Networks

As in the previous section, we will start outlining some of the standard local-based similarity measures for link prediction for bipartite graphs, and some extensions to these, proposed recently. Subsequently we will delineate the maximum entropy approach that we propose.

Let us recall the notation introduced in Chapter 1 for bipartite graphs: we indicate the two layers of the bipartite network as $\top$ and $\perp$; nodes on the layer $\top$ are identified by Latin indices and nodes on the layer $\perp$ with Greek ones. The number of nodes of the two layers is respectively $N_\top$ and $N_\perp$. A bipartite network is described by a bi-adjacency matrix, i.e. the rectangular matrix $\mathbf{M}_{N_\top \times N_\perp}$ whose entries $m_{i\alpha}$ are 1 if there is an edge connecting i and α and 0 otherwise.

Similarity measures for binary bipartite networks

- **Common neighbors (CN)**:

$$\begin{aligned} CN(i,\alpha) &= (N(i) \cap N(N(\alpha))) \cup (N(\alpha) \cap N(N(i))) \\ s_{i\alpha}^{CN} &= |CN(i,\alpha)| \end{aligned} \tag{4.17}$$

  is an index counting the number of paths of length three going from node i to node α plus the number of paths of length three going from node α to node i;

- **Resource Allocation (RA)**:

$$s_{i\alpha}^{RA} = \sum_{z \in CN(i,\alpha)} \frac{1}{|N(z)|}$$

assigns a different weight to the common neighbors of nodes i and α based on its degree;

- **Preferential Attachment (PA)**:

$$s_{i\alpha}^{PA} = k_i \cdot k_\alpha$$

is simply the degree product of nodes i and α, can be used in bipartite networks.

- **Cosine Similarity (CS)**:

$$s_{i\alpha}^{CS} = \frac{s_{i\alpha}^{CN}}{\sqrt{k_i \cdot k_\alpha}}$$

is based on the Cosine distance between two vectors of same length.

In contrast to the existing node-neighborhood-based approaches, the link prediction strategy of other similarity-based models focuses no longer only on groups of common nodes and their node neighbours, but also on the organization of the links between them. In those models, the information content related with the CN nodes is complemented with the topological information emerging from the interactions between them. This mathematical reformulation represents the Cannistraci variations[19] of CN, RA and PA respectively renamed Cannistraci-Alanis-Ravasi (CAR), Cannistraci Resource Allocation (CRA) and Cannistraci Preferential Attachment (CPA) and defined in the following way:

- **CAR index**:

$$s_{i\alpha}^{CAR} = s_{i\alpha}^{CN} \cdot s_{i\alpha}^{LCL}$$

- **CRA index**:

$$s_{i\alpha}^{CRA} = \sum_{z \in CN(i,\alpha)} \frac{|\gamma(z)|}{|N(z)|}$$

- **CPA index**:

$$s_{i\alpha}^{CPA} = e_i \cdot e_\alpha + e_i \cdot s_{i\alpha}^{CAR} + e_\alpha \cdot s_{i\alpha}^{CAR} + (s_{i\alpha}^{CAR})^2$$

where $s_{i\alpha}^{LCL}$ counts the links between the common neighbors of nodes i and α, $|\gamma(z)|$ is the number of links of z with the other neighbors of i and α, while $e(i)$ and $e(\alpha)$ are the number of external links respectively of nodes i and α, where external links means links that are directed toward a node that is not in the set of common neighbors of i and α.

Entropy-based approach to bipartite link-prediction

In complete analogy with the monopartite binary case, we apply a null model to reconstruct the network based on the information used as constraints. In this case we constrain the degree-sequence of each layer. The resulting model is the Bipartite Configuration Model. We briefly recall its specification.

$$p_{i\alpha}^{BiCM} = \frac{x_i y_\alpha}{1 + x_i y_\alpha}, \tag{4.18}$$

where $\vec{x}$ is a transformation of the Lagrange multiplier associated to the degree of the nodes in the $\top$ layer, and $\vec{y}$ is the analogous for the $\perp$ layer. That is their value is determined by solving the system of equations

$$\begin{cases} \sum_\alpha p_{i\alpha} = & k_i^* \\ \sum_i p_{i\alpha} = & k_\alpha^*. \end{cases} \tag{4.19}$$

where $\vec{k}^*$ are the values of the node degrees computed on the observed part of the network. The linkage probability (4.18) will play the role of our score function for the prediction task.

Data

For the weighted prediction we use the directed weighted version of the World Trade Web (WTW) and Italian Electronic Market for Interbank Deposits (E-mid) described in the previous chapter. As far as bipartite networks are concerned, we use two different datasets:

- **MovieLens (ML)**: MovieLens[31] datasets were collected by the GroupLens Research Project at the University of Minnesota. This data set consists of 100000 ratings (1-5) from 943 users on 1682 movies. Each user has rated at least 20 movies and is characterized by some demographic information, such as age, job, sex, state and zipcode. The data was collected through the MovieLens web site (`movielens.umn.edu`) during the seven-month period from September 19th, 1997 through April 22nd, 1998. For the set of movies, there is information on the release year, title and genre. Each user can review a movie with a score that ranges from 1 to 5, according to his level of appreciation. We binarize the network by drawing an edge for a user-movie pair if the user has reviewed the movie;

- **Venezuelan Banks and Assets (VBA)**: Bipartite networks of positions that 69 Venezuelan banks hold in 20 asset classes in the period between December 2013 and June 2015. The dataset was firstly presented and analyzed in [42]. The binarized network has an edge between a bank-asset pair, if the position the bank held in the asset class has a value greater than zero at a given timestamp.

The network statistics for these datasets are presented in Table 4.1

Table 4.1: Data description of the MovieLens graph, and the Venezuelan Banks and Assets graph

Graph	Users (Banks)	Items	Nodes	Edges	Avg. Degree	Avg. Degree (Users)	Avg. Degree (Items)
ML	943	1682	2625	100000	76.19	106.04	59.45
VBA[2]	45	20	65	912	28.06	20.27	45.60

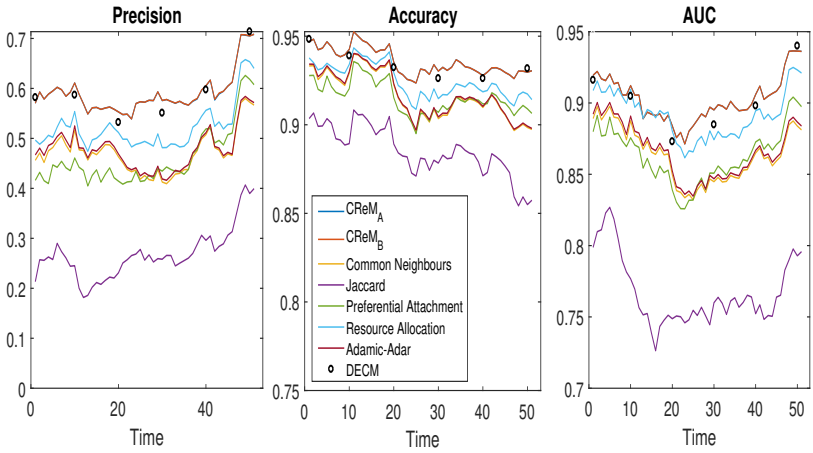

Figure 4.1: Comparison of link-prediction algorithms in the weighted directed case on the WTW dataset. Left panel: evolution of precision; middle panel: evolution of accuracy; right panel: evolution of AUC. The entropy-based approaches to link-prediction perform better than other algorithms, across almost all temporal snapshots.

4.4 Results

Weighted link prediction

Figures 4.1 and 4.2 show the comparison between the topological indicators for the weighted link prediction. We can see that, on both datasets, the performance of the entropy-based methods is superior to the comparison methods on all three indicators. Moreover we observe that CReM$_A$ and CReM$_B$ are coincident, since they use the same linkage probabilities. With few exception, they also perform better than the DECM method. This is remarkable because they have a lower complexity and, CReM$_B$ in particular, is of much faster solution. In light of this result we can recommend the choice of the CReM method over the DECM. As far as the weight imputation is concerned, Figure 4.3 illustrates the goodness of agreement between the weight estimates and the real observed weights. In this case we observe that CReM$_B$ has better performance on

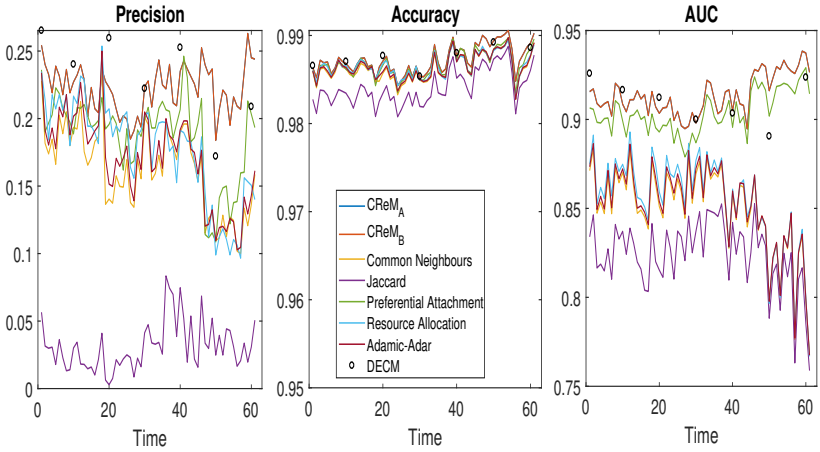

Figure 4.2: Comparison of link-prediction algorithms in the weighted directed case on the E-mid dataset. Left panel: evolution of precision; middle panel: evolution of accuracy; right panel: evolution of AUC. The entropy-based approaches to link-prediction perform better than other algorithms, across almost all temporal snapshots.

both datasets for almost all time snapshots. As a side note, the reason for which we can observe fewer data points for the DECM method is simply because of its elevated complexity, that implies a very slow solution. For comparative purposes we then selected only a subset of the temporal instances. This gives an additional motivation for preferring the CReM methods.

Bipartite link prediction

The results of the algorithm performances are presented in Table 4.2 and Figure 4.4 for the MovieLens data set, and the metrics comparison for the Venezuelan Banks and Assets are in the Figure 4.5.
The results of the link prediction in bipartite networks are averaged over 10 iterations for each method. Table 4.2 shows the averaged measure and the standard deviation (SD).
The results for our entropy based algorithm (BiCM) are comparable and

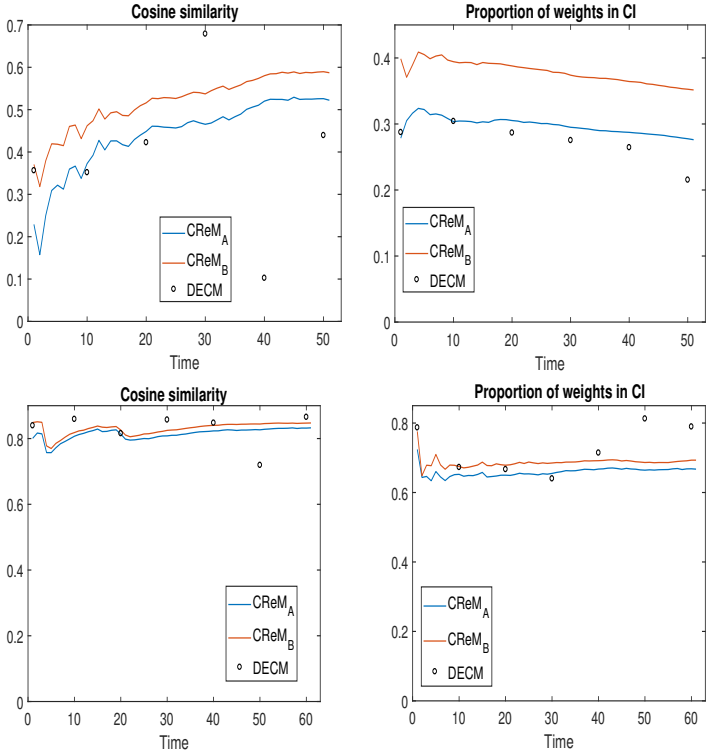

Figure 4.3: Comparison of weights estimations on the WTW (top) and E-mid (bottom) datasets. Left panel: evolution of cosine similarity; right panel: evolution of the proportion of real weights falling in the confidence interval of their estimates. The CReM$_B$ model has a comparable or batter performance on all time snapshots.

show strong performance as opposed to the other algorithms. For the Venezuelan Banks and Assets networks, BiCM link prediction algorithm is among the leaders for the precision and accuracy measures (see Figure 4.5 for multiple time periods). Furthermore, inspecting the Figure 4.5 it can be observed that BiCM method dominates the others in AUC measure. For the MovieLens network BiCM comes in third place for the accuracy (ACC) measure and is the fourth best algorithm for the preci-

Table 4.2: Results: MovieLens performance comparison

Method	ACC (SD)	TPR (SD)	AUC (SD)
BiCM	0.98873 (0.00003)	0.15658 (0.002)	0.8946 (0.002)
cosine	0.98836 (0.00005)	0.12905 (0.004)	0.8903 (0.0007)
car	**0.98917** (0.00003)	0.18975 (0.003)	0.9028 (0.001)
CN	0.98860 (0.00003)	0.14713 (0.002)	0.8868 (0.0009)
cpa	**0.98917** (0.00003)	0.18975 (0.003)	0.9028 (0.001)
cra	0.98929 (0.00003)	**0.19856** (0.002)	**0.9163** (0.001)
PA	0.98873 (0.00003)	0.15672 (0.002)	0.8932 (0.001)
RA	0.98793 (0.00004)	0.09712 (0.003)	0.8863 (0.0007)

sion and AUC (see Table 4.2) and is closely trailing the best performers.

It is important to observe that although the BiCM is not the best performer in absolute terms, it achieves comparable performances with respect to the alternatives making use of a considerably smaller amount of information. In fact, it only requires the knowledge of the degree sequences of the two layers, while other comparison methods require at least to run over all the common neighbours and at most to account for how interconnected those neighbours are. The only method that takes as input the same information, the Preferential Attachment, shows a considerably poorer performance, implying that the BiCM makes an optimal use of the input information.

4.5 Discussion

This chapter presented two different ways to enrich the range of applicability of the entropy-based approach for link prediction: we considered the case of weighted and bipartite networks.

The proposed methods significantly enrich the existing literature. When considering weighted networks, the vast majority of algorithms only allows for missing link prediction disregarding the issue of estimating the weights to be attributed to the links. Among the few methods that allow for weight estimation, some have hidden parameters that are tuned a posteriori [75], while others, using the weights estimates themselves

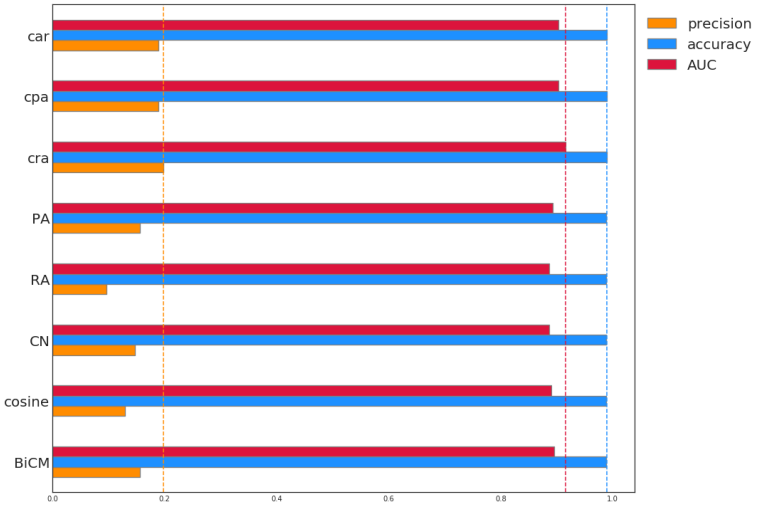

Figure 4.4: Link prediction performance on the MovieLens dataset, a bipartite graph of users linked to movies they reviewed. The BiCM method performs on par with the alternative methods on all evaluation measures.

as scores for the binary prediction, de facto assume that the topology of the network is implied in the weight estimates [49]. Here we proposed a general framework that allows to rigorously estimate the most likely missing links and their weights in a probabilistic way. We also provide a way to compute confidence intervals for the missing link. We proposed three different null models to be used for the task, and we observed that the one that was performing best on most cases was also the one of lowest algorithmic complexity. We therefore concluded that such method, based on the CReM$_B$ null model, is preferable over the alternatives. We will soon release a routine for link prediction via CReM$_B$ to make its application straightforward also for non-experts. One more observation is due on the different performances of DECM and CReM with respect to the topological estimation. The DECM incorporates in the binary linkage probabilities the Lagrange multipliers relative to both the degree and the strength sequence, while CReM uses the DBCM linkage probabilities, that only impose the preservation of the degrees. In other

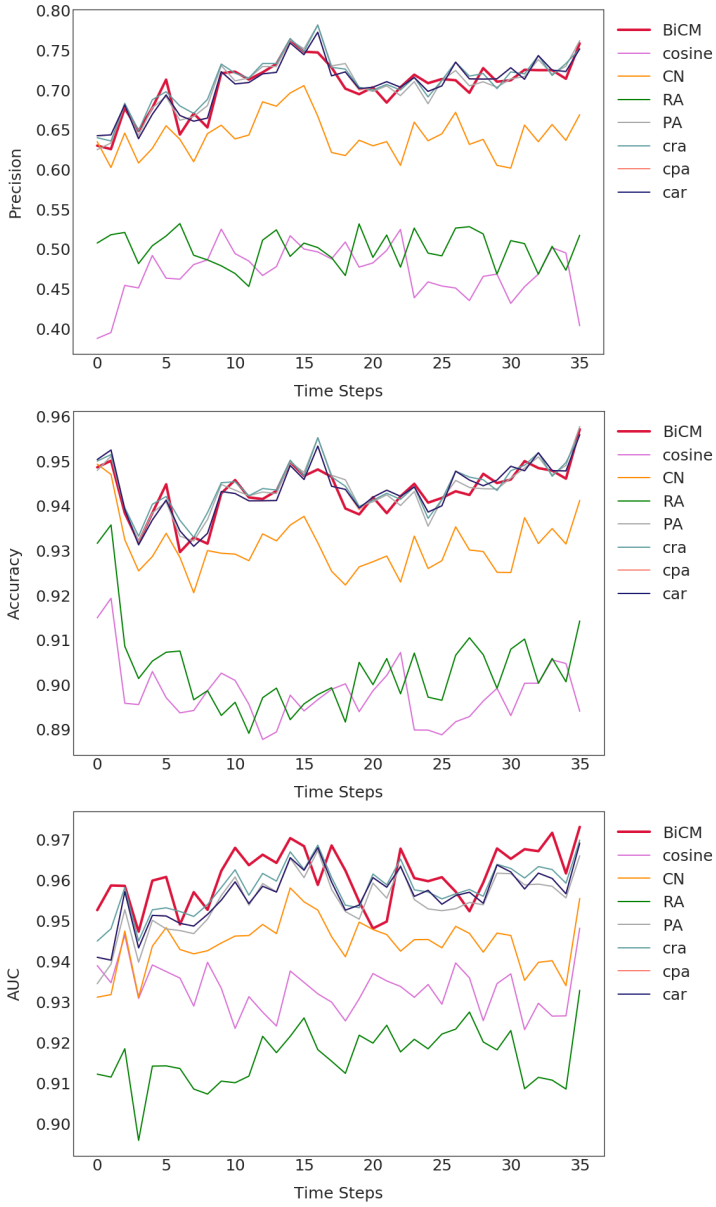

Figure 4.5: Link prediction performance on the Venezuelan Banks and Assets dataset. Performance is measured by Precision (top), Accuracy (middle) and AUC (bottom).

words, the DECM includes the weighted information also in the binary linkage probability, while the CReM does not. As we can see from Figures 4.1 and 4.2, there is not a clear winner between this two different approaches. This confirms other studies in the literature that showed that adding weight information in the topology prediction phase can sometimes improve and sometimes worsen the performance.

On the front of bipartite network we presented an extension to the method of the previous chapter based on the Bipartite Configuration Model. This methods enriches the literature on bipartite prediction. The prediction results show good (even if not best) performances with respect to the alternative methods. It is to be noted, though, that all other methods use significantly more information, relying on the exploration of all the common neighbours of two nodes, or even the number of link connecting such common neighbours. The BiCM approach on the other hand only requires the degree sequence of each layer, making it a better candidate especially for dense datasets, where the exploration of all common neighbours and their interconnectivity can result to be very costly,if not completely infeasible, in terms of calculation time and memory required. The only method that uses the same amount of information as the BiCM, Preferential Attachment, shows a significantly poorer performance, suggesting that our method achieves an optimal use of the input information.

Chapter 5

Conclusion

All good things come to an end, and thankfully, also all other things. Before leaving you to your happily ever after, let me conclude this work by going through the main achievements and underlying the open paths that still lie ahead of us.
Despite the ever increasing amount of data available for all sorts of aspects of human life and various fields of science, it is very often the case that such data is still incomplete, partially incorrect or inaccessible. Privacy regulations forbid the disclosure of sensitive information, costly experiments slow down the acquisition process, misalignment of international regulations results in delays in data availability, observation and transmission processes cause the data to be partially inexact or incomplete. Statistical studies have underlined how rigorously account for missing information is vital to avoid introducing biases in the results. Therefore many scholars have devoted their efforts in the creation of procedures to perform such tasks. In the particular case of complex networks, the literature is rich with methods devised to perform both tasks we have focused on: *network reconstruction* and *link imputation*. Yet, as all authors before us likely did, we claim that our solutions fill a methodological gap in the literature. If you are still not convinced of this, besides questioning my explanatory abilities, I should try to clarify our contributions.

In the network reconstruction case, we focus on weighted networks. We observed that the works proposed in the literature follow two main lines: in some cases, they attribute weights deterministically on top of a binary distribution, thus completely separating the task of reconstructing the network topological structure and its weighted characteristics. In this way not only the interplay between link formation and link intensity is neglected, but the fact that the methods choose a single realization out of a binary distribution and add weights deterministically implies that the likelihood of reproducing any real matrix is null. On the other end of the spectrum, we have methods, like the (D)ECM, that rigorously estimates an unbiased weighted distribution, but assume that both topology and weights are *jointly* determined by binary and weighted constraints. That is, the implicit assumption of the method is that not only the network topology affects the weight distribution, as it is clearly the case, but also the reverse is true: the parameters relative to the weighted constraints appear in the expression of the binary distribution. However, the latter implication does not necessarily hold. In general it is possible that in the network formation process the structure is first determined and then the link intensity adjusted.

Our approach sits in between these two lines of methodology. In fact, we aim at deriving a flexible method that allows all the available topological information to be used, and which determines an unbiased weighted distribution taking into account the binary one. Let us be more specific. We start with some knowledge over the binary distribution of a weighted network. This can be achieved because of prior information or by solving *any* binary model. We suggested two possible choices for such model, depending on the information available, but the user can make any choice. Adding to this prior information some weighted constraints (i.e. the strength sequences) we derived an unbiased weighted distribution *conditional* on the binary one. In this way we achieve several desirable characteristics: flexibility in the choice of the binary model, allowing for different types of binary information to be used; derivation of an unbiased weighted distribution based upon the binary one. As a consequence the likelihood of observing the real network is not null.

Moreover, in the second specification of the model, CReM$_B$, the method is efficient and fast since it does not require the solution of systems of coupled equations, like the DECM does. From a methodological point of view, we applied the maximum entropy framework but with an important difference: the key quantity to be maximized is now the *conditional entropy* of the weighted distribution given the binary one. In order to perform the parameter estimation part, we proposed a new functional, the *generalized likelihood*, that, as the name suggests, extend the concept of likelihood by averaging over the space of all possible binary configurations. The comparison we carried out with two different methodologies shows that the Conditional Reconstruction Method has better or comparable performances with respect to the ECM (which had been shown to have very good performances in the reconstruction task), but with a significantly lowered the complexity.

As far as the link imputation problem is concerned, our approach relies on an idea that is different in spirit from many methods proposed in the literature. We move from the concept of reconstructability of a network: if we are able to find a method that, given some observables, is able to satisfactorily reconstruct the network structure, than such model will also be able to deliver meaningful insights on the unobserved part of a partially observed network. A few observations are due: first of all, the reader might remember that, when introducing the concept of network ensemble and the ERG model, we pointed out that the choice of using a canonical ensemble (as opposed to the micro-canonical), besides simplifying the mathematical tractability of the problem, was also fundamental because it allowed perturbations of the network observables. In the case of link imputation this is a strict necessity: since we only have access to a part of the network, any quantity measured on it will be perturbed with respect to the "real" value relative to the underlying hidden network. When using reconstruction methods for link imputation we therefore rely on a few assumptions: the network observable we selected is "explanatory enough" to allow for a satisfactory reconstruction of the network characteristics; the network is "homogeneous enough" that the information on the observed part is also meaningful with respect

to the unobserved one and, consequently, that the perturbation of the observables given by the incomplete information is "small enough" not to biased the reconstruction procedure. With regard to the first assumption, we are now in a better position with respect to the network reconstruction task, since we can select virtually any observable: the network structure and its characteristics are known, although only partially. The second assumption implies that we assume the unobserved part of the network not to be structurally different from the observed one. Whether or not this assumption holds depends on the reason for which part of the network is not observed. Given this observations, it is expectable that the proposed method might fail when the reconstruction is not successful. For instance, in the case of networks with a strong community structure. In this case, in fact, a null model based on the knowledge of degree and strength sequence does not contain sufficient information of the community structure and therefore fails to capture structural properties of the graph. However, the proposed framework is flexible and allows the use of virtually any null model of choice. If the linkage probabilities are computed by solving a model that accounts for communities, the results might improve. This represents a possible future extension of the current work.

Still on the imputation problem, we presented two extensions: the one relative to bipartite graphs represented an interesting exercise, contributing to a literature that is much less flourishing than its monopartite counterpart. The extensions to weighted networks, on the other hand, is particularly interesting because it contributes to a problem that is strongly under-represented in existing works: the estimation of link weights. Almost all methods that account for weighted information in the link imputation procedure do not propose a procedure to infer the intensity of the link weights. As we previously noted, the only two methods, to the best of our knowledge, that allow such estimation need to be take with care. In one case [75], the authors estimate the model parameter on the inaccessible portion of the data, thus making the method concretely inapplicable in real missing link problems. The second paper [49], presents an interesting and fairly complicated methodology, targeted to the case

of social networks. The main drawback of this approach is given by the fact that it uses the weights estimates themselves as scores for the imputation of the missing links. This corresponds to a strong assumption with respect to the interplay of topology and weighted structure, as we discussed at length in Chapter 2. Therefore, our method presents a meaningful advancement with respect to state of the art methods. Moreover, since the use of reconstruction methods returns not only linkage probabilities and expected weights, but the complete specification of binary and weighted distributions, we are able to compute confidence intervals for the weight estimates.

In conclusion, the work presented in this thesis addresses some interesting issues but also opens the door to more questions. It remains to be investigated how to estimate the proportion of a missing links given a partially observed network, or how to account for the presence of spurious links. The latter problem might seem like a trivial extension, but it is actually quite subtle. I hope we will be able to address these and other issues in the future so that we can say: "the best is yet to come".

Bibliography

[1] Lada A Adamic and Eytan Adar. "Friends and neighbors on the web". In: *Social networks* 25.3 (2003), pp. 211–230.

[2] Hirotugu Akaike. "A new look at the statistical model identification". In: *IEEE transactions on automatic control* 19.6 (1974), pp. 716–723.

[3] M Andrecut. "Systemic risk, maximum entropy and interbank contagion". In: *International Journal of Modern Physics C* 27.12 (2016), p. 1650148.

[4] Sandro Azaele et al. "Statistical mechanics of ecological systems: Neutral theory and beyond". In: *Rev. Mod. Phys.* 88.3 (2016). ISSN: 15390756. DOI: 10.1103/RevModPhys.88.035003. arXiv: 1506.01721v1.

[5] Michael Bacharach. "Estimating nonnegative matrices from marginal data". In: *International Economic Review* 6.3 (1965), pp. 294–310.

[6] M Baltakiene et al. "Maximum entropy approach to link prediction in bipartite networks". In: *arXiv preprint arXiv:1805.04307* (2018).

[7] Albert-László Barabási and Réka Albert. "Emergence of scaling in random networks". In: *science* 286.5439 (1999), pp. 509–512.

[8] Baruch Barzel and Albert-László Barabási. "Network link prediction by global silencing of indirect correlations". In: *Nature biotechnology* 31.8 (2013), p. 720.

[9] C. Becatti, G. Caldarelli, and F. Saracco. "Entropy-based randomisation of rating networks". In: *ArXiv e-prints* (May 2018). arXiv: 1805.00717 [physics.soc-ph].

[10] Giulia Berlusconi et al. "Link prediction in criminal networks: A tool for criminal intelligence analysis". In: *PloS one* 11.4 (2016), e0154244.

[11] Guido Caldarelli. *Scale-free networks: complex webs in nature and technology*. Oxford University Press, 2007.

[12] Carlo Vittorio Cannistraci, Gregorio Alanis-Lobato, and Timothy Ravasi. "From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks". In: *Scientific reports* 3 (2013), p. 1613.

[13] Giulio Cimini et al. "Estimating topological properties of weighted networks from limited information". In: *Physical Review E* 92.4 (2015), p. 040802.

[14] Giulio Cimini et al. "Systemic risk analysis on reconstructed economic and financial networks". In: *Scientific reports* 5 (2015), p. 15758.

[15] Giulio Cimini et al. "The Grand Canonical Ensemble of Weighted Networks". In: *in preparation* (2018).

[16] Giulio Cimini et al. "The Statistical Physics of Real-World Networks". In: *arXiv preprint arXiv:1810.05095* (2018).

[17] Thomas M Cover and Joy A Thomas. *Elements of Information Theory*. 2006.

[18] Matthieu Cristelli et al. "Measuring the Intangibles: A Metrics for the Economic Complexity of Countries and Products". In: *PLOS ONE* 8.8 (2013).

[19] Simone Daminelli et al. "Common neighbours and the local-community-paradigm for topological link prediction in bipartite networks". In: *New J. Phys.* 17.11 (2015). ISSN: 13672630. DOI: 10.1088/1367-2630/17/11/113037. arXiv: 1504.07011.

[20] C F Dormann et al. "Indices, graphs and null models: analysing bipartite ecological networks". In: *Open Ecol. J.* 2 (2009), pp. 7–24. ISSN: 18742130. DOI: 10.2174/1874213000902010007.

[21] Agata Fronczak. "Exponential random graph models". In: *Encyclopedia of Social Network Analysis and Mining* (2014), pp. 500–517.

[22] Axel Gandy and Luitgard AM Veraart. "A Bayesian methodology for systemic risk assessment in financial networks". In: *Management Science* 63.12 (2016), pp. 4428–4446.

[23] Man Gao et al. "Projection-based link prediction in a bipartite network". In: *Information Sciences* 376 (2017), pp. 158–171.

[24] Diego Garlaschelli and Maria I Loffredo. “Generalized bose-fermi statistics and structural correlations in weighted networks”. In: *Physical review letters* 102.3 (2009), p. 038701.

[25] Diego Garlaschelli and Maria I Loffredo. “Maximum likelihood: Extracting unbiased information from complex networks”. In: *Physical Review E* 78.1 (2008), p. 015101.

[26] Kristian Skrede Gleditsch. “Expanded trade and GDP data”. In: *Journal of Conflict Resolution* 46.5 (2002), pp. 712–724.

[27] Stanislao Gualdi et al. “Statistically validated network of portfolio overlaps and systemic risk”. In: *Sci. Rep.* 6 (Mar. 2016). ISSN: 20452322. DOI: 10.1038/srep39467. arXiv: 1603.05914. URL: http://arxiv.org/abs/1603.05914%20http://dx.doi.org/10.1038/srep39467.

[28] Jean-Loup Guillaume and Matthieu Latapy. “Bipartite structure of all complex networks”. In: *Inf. Process. Lett.* 90.5 (2004), pp. 215–221. ISSN: 0020-0190. DOI: 10.1016/j.ipl.2004.03.007. URL: http://www.sciencedirect.com/science/article/B6V0F-4C6KTH0-1/2/f86fa29c9f27bd046fc0e06473dc1a07%7B%5C$%7D%7B%5C%%7D5C%7B%5C$%7Dnhttp://www.sciencedirect.com/science?%7B%5C%%7D7B%7B%5C_%7D%7B%5C%%7D7Dob=ArticleURL%7B%5C%%7D7B%7B%5C&%7D%7B%5C%%7D7D%7B%5C%%7D7B%7B%5C_%7D%7B%5C%%7D7Dudi=B6V0F-4C6KTH0-1%7B%5C%%7D7B%7B%5C&%7D%7B%5C%%7D7D%7B%5C%%7D7B%7B%5C_%7D%7B%5C%%7D7Duser=2400262%7B%5C%%7D7B%7B%5C&%7D%7B%5C%%7D7D%7B%5C%%7D7B%7B%5C_%7D%7B%5C%%7D7Drdoc=1%7B%5C%%7D7B%7B%5C&%7D%7B%5C%%7D7D%7B%5C%%7D7B%7B%5C_%7D%7B%5C%%7D7Dfmt=.

[29] Roger Guimerà and Marta Sales-Pardo. “Missing and spurious interactions and the reconstruction of complex networks”. In: *Proceedings of the National Academy of Sciences* 106.52 (2009), pp. 22073–22078.

[30] Grzegorz Hałaj and Christoffer Kok. “Assessing interbank contagion using simulated networks”. In: *Computational Management Science* 10.2-3 (2013), pp. 157–186.

[31] F. Maxwell Harper and Joseph A. Konstan. “The MovieLens Datasets: History and Context”. In: *ACM Transactions on Interactive Intelligent Systems (TiiS)* 4.6 (2015). DOI: 10.1145/2827872.

[32] César A Hidalgo and Ricardo Hausmann. "The building blocks of economic complexity." In: *Proc. Natl. Acad. Sci. U. S. A.* 106.26 (June 2009), pp. 10570–10575. ISSN: 1091-6490. DOI: 10.1073/pnas.0900943106. URL: http://www.pnas.org/cgi/content/long/106/26/10570.

[33] Giulia Iori, Saqib Jafarey, and Francisco G Padilla. "Systemic risk on the interbank market". In: *Journal of Economic Behavior & Organization* 61.4 (2006), pp. 525–542.

[34] Paul Jaccard. "Étude comparative de la distribution florale dans une portion des Alpes et des Jura". In: *Bull Soc Vaudoise Sci Nat* 37 (1901), pp. 547–579.

[35] Mahdi Jalili et al. "Link prediction in multiplex online social networks". In: *Royal Society open science* 4.2 (2017), p. 160863.

[36] Edwin T Jaynes. "Information theory and statistical mechanics". In: *Physical review* 106.4 (1957), p. 620.

[37] Edwin T Jaynes. "On the rationale of maximum-entropy methods". In: *Proceedings of the IEEE* 70.9 (1982), pp. 939–952.

[38] Leo Katz. "A new status index derived from sociometric analysis". In: *Psychometrika* 18.1 (1953), pp. 39–43.

[39] Jérôme Kunegis and Andreas Lommatzsch. "Learning spectral graph transformations for link prediction". In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM. 2009, pp. 561–568.

[40] Matthieu Latapy, Clémence Magnien, and Nathalie Del Vecchio. "Basic notions for the analysis of large two-mode networks". In: *Soc. Networks* (2008). ISSN: 03788733. DOI: 10.1016/j.socnet.2007.04.006. arXiv: 0611631 [cond-mat].

[41] Iman van Lelyveld et al. "Finding the core: Network structure in interbank markets". In: *Journal of Banking & Finance* 49 (2014), pp. 27–40.

[42] Sary Levy-Carciente et al. "Dynamical macroprudential stress testing using network theory". In: *J. Bank. Financ.* 59 (2015), pp. 164–181. ISSN: 03784266. DOI: 10.1016/j.jbankfin.2015.05.008.

[43] Hao Liao, An Zeng, and Yi-Cheng Zhang. "Predicting missing links via correlation between nodes". In: *Physica A: Statistical Mechanics and its Applications* 436 (2015), pp. 216–223.

[44] David Liben-Nowell and Jon Kleinberg. "The Link-Prediction Problem for Social Networks". In: *Conference on Information and Knowledge Management (CIKM'03)*. 2003, pp. 556–559.

[45] Greg Linden, Brent Smith, and Jeremy York. "Amazon.com Recommendations". In: *IEEE Internet Comput.* February (2003), pp. 76–80. ISSN: 1089-7801. DOI: 10.1109/MIC.2003.1167344.

[46] Linyuan Lü and Tao Zhou. "Link prediction in complex networks: A survey". In: *Physica A: statistical mechanics and its applications* 390.6 (2011), pp. 1150–1170.

[47] Linyuan Lü et al. "Toward link predictability of complex networks". In: *Proceedings of the National Academy of Sciences* 112.8 (2015), pp. 2325–2330.

[48] Rossana Mastrandrea et al. "Enhanced reconstruction of weighted networks from strengths and degrees". In: *New Journal of Physics* 16.4 (2014), p. 043022.

[49] Behnaz Moradabadi and Mohammad Reza Meybodi. "Link prediction in weighted social networks using learning automata". In: *Engineering Applications of Artificial Intelligence* 70 (2018), pp. 16–24.

[50] M. E. J. Newman. "The structure of scientific collaboration networks". In: *Proc. Natl. Acad. Sci. U. S. A.* 98.2 (2001), pp. 404–409. ISSN: 0027-8424. DOI: 10.1073/pnas.021544898. arXiv: 0007214 [cond-mat]. URL: http://www.pnas.org/content/98/2/404.abstract.

[51] Liming Pan et al. "Predicting missing links and identifying spurious links via likelihood analysis". In: *Scientific reports* 6 (2016), p. 22955.

[52] Federica Parisi, Guido Caldarelli, and Tiziano Squartini. "Entropy-based approach to missing-links prediction". In: *Applied Network Science* 3.17 (2018).

[53] Federica Parisi, Tiziano Squartini, and Diego Garlaschelli. "A faster horse on a safer trail: generalized inference for the efficient reconstruction of weighted networks". In: *arXiv preprint arXiv:1811.09829* (2018). URL: https://arxiv.org/abs/1811.09829.

[54] Juyong Park and Mark EJ Newman. "Statistical mechanics of networks". In: *Physical Review E* 70.6 (2004), p. 066117.

[55] Milen Pavlov and Ryutaro Ichise. "Finding experts by link prediction in co-authorship networks." In: *FEWS* 290 (2007), pp. 42–55.

[56] Hially Rodrigues Sa and Ricardo BC Prudencio. "Supervised learning for link prediction in weighted networks". In: *III International Workshop on Web and Text Intelligence*. 2010.

[57] Gerard Salton and Michael J McGill. "Introduction to modern information retrieval". In: (1983).

[58] Fabio Saracco et al. "Inferring monopartite projections of bipartite networks: An entropy-based approach". In: *New J. Phys.* 19.5 (July 2017), p. 16. ISSN: 13672630. DOI: 10.1088/1367-2630/aa6b38. arXiv: 1607.02481. URL: http://arxiv.org/abs/1607.02481.

[59] Daniel Schall. "Link prediction in directed social networks". In: *Social Network Analysis and Mining* 4.1 (2014), p. 157.

[60] Kushal Veer Singh and Lovekesh Vig. "Improved prediction of missing protein interactome links via anomaly detection". In: *Applied Network Science* 2.1 (2017), p. 2.

[61] Th A Sorensen. "A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons". In: *Biol. Skar.* 5 (1948), pp. 1–34.

[62] Tiziano Squartini, Giorgio Fagiolo, and Diego Garlaschelli. "Randomizing world trade. I. A binary network analysis". In: *Physical Review E* 84.4 (2011), p. 046117.

[63] Tiziano Squartini and Diego Garlaschelli. "Analytical maximum-likelihood method to detect patterns in real networks". In: *New Journal of Physics* 13.8 (2011), p. 083001.

[64] Tiziano Squartini and Diego Garlaschelli. *Maximum-Entropy Networks: Pattern Detection, Network Reconstruction and Graph Combinatorics*. Springer, 2017.

[65] Tiziano Squartini and Diego Garlaschelli. "Reconnecting statistical physics and combinatorics beyond ensemble equivalence". In: *arXiv preprint arXiv:1710.11422* (2017).

[66] Tiziano Squartini et al. "Network reconstruction via density sampling". In: *Applied Network Science* 2.1 (2017), p. 3.

[67] Tiziano Squartini et al. "Reconstruction methods for networks: the case of economic and financial systems". In: *arXiv preprint arXiv:1806.06941* (2018).

[68] Samir Suweis et al. “Emergence of structural and dynamical properties of ecological mutualistic networks”. In: *Nature* 500.7463 (2013), pp. 449–52. ISSN: 1476-4687. DOI: 10.1038/nature12438. URL: http://www.ncbi.nlm.nih.gov/pubmed/23969462.

[69] Fei Tan, Yongxiang Xia, and Boyao Zhu. “Link prediction in complex networks: a mutual information perspective”. In: *PloS one* 9.9 (2014), e107056.

[70] Christian Upper. “Simulation methods to assess the danger of contagion in interbank markets”. In: *Journal of Financial Stability* 7.3 (2011), pp. 111–125.

[71] Simon J Wells. “Financial interlinkages in the United Kingdom’s interbank market and the risk of contagion”. In: (2004).

[72] JieHua Wu et al. “Weighted Local Naive Bayes Link Prediction.” In: *Journal of Information Processing Systems* 13.4 (2017).

[73] Zhongqi Xu et al. “Entropy-based link prediction in weighted networks”. In: *Chinese Physics B* 26.1 (2017), p. 018902.

[74] Muhammed A Yildirim and Michele Coscia. “Using random walks to generate associations between objects”. In: *PloS one* 9.8 (2014), e104813.

[75] Jing Zhao et al. “Prediction of links and weights in networks by reliable routes”. In: *Scientific reports* 5 (2015), p. 12261.

[76] Tao Zhou, Linyuan Lü, and Yi-Cheng Zhang. “Predicting missing links via local information”. In: *The European Physical Journal B* 71.4 (2009), pp. 623–630.

[77] Boyao Zhu and Yongxiang Xia. “Link prediction in weighted networks: A weighted mutual information model”. In: *PloS one* 11.2 (2016), e0148265.