Inference and Information in Network Structure

by

Maximilian Jerdee

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Physics)
in the University of Michigan
2025

Doctoral Committee:

Professor Mark Newman, Chair
Associate Professor Elizabeth Bruch
Assistant Professor Abigail Z. Jacobs
Professor Cagliyan Kurdak
Professor Xiaoming Mao

Maximilian Jerdee
mjerdee@umich.edu
ORCID iD: 0009-0005-2268-5412

Dedication

To Mom & Dad

Acknowledgements

I would like to express my sincere gratitude to all those who have supported me throughout the completion of this thesis. My doctoral journey has been an immensely fulfilling experience, only made possible by a tremendous network of support.

First and foremost, I am thankful to my advisor, Mark Newman, for his expertise, encouragement, and guidance. His insightful feedback and constant engagement have been crucial to the development of this research and my scientific growth. I am fortunate to have learned from such a dedicated mentor.

I extend heartfelt thanks to my thesis committee for their valuable time and effort in reviewing my work, which has wound up more long-winded than I anticipated.

My appreciation also goes to the broader physics department and its staff for creating a supportive environment for everyone. Their efforts to include and enable graduate students to engage with the department have considerably enriched my time at Michigan. I am particularly grateful for support from Leopoldo Pando Zayas and the LCTP group.

I am also glad to have been embraced by the Complex Systems community, whose endless stream of conversations and snacks have fostered a wonderful academic home. Special thanks to my group-mate Austin Polanco, whose advice and LaTeX template have been instrumental in completing this thesis.

To my hundreds of housemates at the Ella Baker Co-operative over the years: thank you for being such a welcoming community and enabling me to pursue my cooking ambitions. I am especially grateful to Vellia Zhou for her love and motivation.

I would like to acknowledge the teachers and mentors who guided my academic path from science fairs to math teams. Their encouragement laid the foundation for my pursuit of higher education and inspires me to live up to their generosity.

Lastly, I am deeply grateful to my family—Mom, Dad, Olivia, and Alex—for their unwavering support throughout my educational journey. Being away from home in New Jersey during my PhD has been challenging, but their love has never felt far.

This thesis reflects not only my work but also the support and guidance of everyone mentioned here and so many others, and I am truly grateful to have each of you in my life.

Preface

This dissertation is based upon research done in physics and network science over the course of my PhD.

The Introduction is meant as a pedagogical review of topics in network science that could be used as an entry-point to the field. This extended background aims to convey a perspective on modeling and understanding systems applicable to a far broader set of applications than the network science of this thesis. Most of the information presented is standard material in the field although some topics are original unpublished work, for instance the general configuration model of Section 1.2.2.

Chapter 2 presents two unpublished projects on network group structure I have independently pursued. Chapter 3 is based upon two information-theoretic endeavors completed in collaboration with Mark Newman and Alec Kirkley. The first, Section 3.1, is currently under review [66] while the second, Section 3.2, has been published [65]. Chapter 4 is based upon published work on hierarchies with Mark Newman [67].

A significant amount of additional material is presented in the Appendices. Appendix A supplements the Introduction with self-contained reviews of the fields of statistics, physics, and information theory relevant to my work. Appendices B, C, and D provide details supporting Chapters 2, 3, and 4 respectively.

Table of Contents

Dedication ii

Acknowledgements iii

Preface iv

List of Figures vii

List of Tables xviii

List of Appendices xix

Abstract xxi

Chapter 1. Introduction **1**

1.1 Networks and complex systems 1
1.1.1 Representing network data 1
1.1.2 Group structure 4
1.1.3 Hierarchy structure 7
1.2 Random graph models 9
1.2.1 Erdős-Rényi graph model 10
1.2.2 Configuration model 19
1.2.3 Stochastic block model 29
1.2.4 Bradley-Terry model 39
1.3 Contributions 45

Chapter 2. Group Structure **47**

2.1 Degree-correction of the stochastic block model 47
2.1.1 Introduction 47
2.1.2 Traditional degree-correction 49
2.1.3 General degree-correction 54
2.1.4 Results 56
2.1.5 Conclusions 60
2.2 Assortative stochastic block models 60
2.2.1 Introduction 60
2.2.2 Resolution limit 62
2.2.3 Simple assortative SBM 64

2.2.4 General assortative SBM 65
2.2.5 Synthetic tests 70
2.2.6 Results 72
2.2.7 Conclusions 78

Chapter 3. Information Theory and Clustering Similarity 79

3.1 Mutual information of clusterings 79
3.1.1 Introduction 80
3.1.2 Conditional entropy and mutual information 81
3.1.3 Example applications 96
3.1.4 Conclusions 102
3.2 Normalized mutual information 104
3.2.1 Introduction 104
3.2.2 Normalization of the mutual information 106
3.2.3 Example application: Community detection 112
3.2.4 Comparison between variants of the mutual information 113
3.2.5 Comparison between community detection algorithms 116
3.2.6 Conclusions 116

Chapter 4. Hierarchies 118

4.1 Introduction 118
4.2 The model 121
4.3 Extensions of the model 123
4.3.1 Upset wins and luck 123
4.3.2 Depth of competition 124
4.3.3 Combined model 126
4.3.4 Minimum violations ranking 127
4.4 Results 128
4.4.1 Luck and parameter identifiability 131
4.5 Predicting wins and losses 132
4.6 Conclusion 135

Chapter 5. Conclusion 138

Appendices 141

Bibliography 229

List of Figures

1.1 Network of interactions observed by Setia *et al.* [120] among $n = 6$ orangutans represented (a) as a graph and (b) as an adjacency matrix $\boldsymbol{A}$. (c) The node degrees $\boldsymbol{k}$, the number of interactions of each orangutan. 1

1.2 Directed multigraph of dominance interactions observed among $n = 6$ orangutans [120] represented (a) as a graph and (b) as an adjacency matrix $\boldsymbol{A}$. (c) The out-degrees $\boldsymbol{k}^{\text{out}}$ and (d) the in-degrees $\boldsymbol{k}^{\text{in}}$ are also given, respectively counting the number of times each orangutan exhibits a dominant or submissive interaction. 3

1.3 Four examples of community structure in networks. (a) American football matches among Division IA colleges in the 2000 regular season [57]. Most matches occur within the labeled "conferences" that the teams belong to. (b) Network of political books frequently co-purchased on Amazon.com near the 2004 U.S. presidential election labeled by political lean. (c) Network of characters in Shakespeare's *Romeo and Juliet* where edge thickness indicates frequency of interaction. The house each character belongs to in the work, Montague or Capulet, is highlighted. (d) Bipartite network of plant-pollinator interactions [16] 5

1.4 Four examples of hierarchy structure in networks. (a) American football matches played in the 2022 Big Ten conference football season [2]. The hierarchical structure is represented vertically, with better teams located higher. Arrows run from the winner to the loser of a match, colored green if the higher ranked team wins, red if the lower ranked team wins. (b) Consumer preferences between pairs of sodas [46]. (c) Observed dominance interactions (pecks) among 11 chickens [87]. (d) Unreciprocated friendship nominations among 20 high school students [128]. 7

1.5 (a) A social network of friendships among Facebook profiles [17]. A path of 6 edges, highlighted in red, connecting the leftmost and rightmost nodes. (b) Posterior distribution $P(p|\boldsymbol{A})$ of the connection probability p for this network in the Erdős-Rényi model. The MAP estimate $\hat{p}_{\text{MAP}} \approx 0.042$ is marked by the vertical line. 12

1.6 Samples of an Erdős-Rényi random graph on $n = 50$ vertices at edge probabilities $p = 0.004, 0.02, 0.1$ corresponding to average degrees $c = 0.2, 1, 5$. Nodes are colored by the connected component they belong to. The middle panel shows the emergence of the giant component at the transition point $c = 1$. . . . 13

1.7 (a) Social network of Facebook profile connections, nodes sized according to their degree [17]. (b) A sampled Erdős-Rényi random graph with the same number of nodes and edges as the social network. (c) Histogram of degree distributions of the social network (blue) and random graph (orange). (d) Distribution of the maximum degree k_{max} in the random graph null model, as observed over 1,000 samples. The maximum degree of 58 achieved by the social network is highlighted in blue and is well-separated from that obtained by a typical random graph. 20

1.8 Schematic of perspectives on the generative process in the configuration model. (a) Canonically, the overall density ρ is first sampled from the prior $P(\rho)$, followed by the node weights $\boldsymbol{\theta}$ indicated by the node sizes, and then the network $\boldsymbol{A}$ itself is generated with edges appearing proportionally to the weights they connect. Example parameters ρ, θ_1, θ_2 relevant to generating A_{12} are given. (b) Microcanonically, the number of edges m is first set, followed by a degree sequence $\boldsymbol{k}$ that sums to $2m$. Finally the network is generated by a stub matching process that wires together the edges. 22

1.9 (a) Degree distributions of the Facebook friend and football match networks. (b) Posterior distributions of the degree homogeneity parameter α in the general configuration model on these networks. Values of α are plotted along the bottom x-axis and corresponding values of the Gini coefficient of degree inequality G are given on the top x-axis. The overall scale is linear in the Gini coefficient, leading to a Jacobian factor in the posterior such that the plot reflects the appropriate density. The configuration model is highlighted by the vertical line at $\alpha = 1, G = 0.5$, along with the Erdős-Rényi model at $\alpha \to \infty, G = 0$. 25

1.10 (a) Football match network with edges within groups highlighted in green and edges between groups in red. (b) Modularity of the conference partition of the football match network (vertical dashed line) compared to 10,000 alternative networks with the same degree sequence sampled from the configuration model. 31

1.11 (a) Network of football matches within only the Big Ten conference. The teams are partitioned into two groups to maximize modularity. (b) The modularity along the optimized partition compared that in 10,000 networks with the same degree sequence sampled from the configuration model. 32

1.12 Example generative process for the stochastic block model of group structure. The group each node belongs to $\boldsymbol{b}$, represented by the colors, is first sampled. The overall density ρ is then sampled and used to generate the symmetric weight matrix ω. In the final network $\boldsymbol{A}$ the density of edges between a node in group r and a node in group s is given by the entry ω_{rs}. In this example the edges within groups (colored green) are much more likely than edges between groups (red), resulting in an assortative group structure. 36

1.13 Posterior distributions of the number of groups q found by the stochastic block model within the networks of football matches within only the Big Ten conference (orange) and the entire Division IA (blue). The special case of $q = 1$, for which the SBM reduced to the Erdős-Rényi random graph is highlighted in red. We observe that while there is considerable evidence of group structure in the Division IA network, likely into 10 groups. The Big Ten conference, however, does not meaningfully have internal group structure. 38

1.14 College football matches played in the 2022 Big Ten conference football season. (a) Arrows represent that one team beat another, teams are vertically arranged to minimize the number of upset victories (in red). (b) Arrows indicate that one team hosts another. Teams are vertically positioned to minimize ranking violations. (c) The number of times the favorite "wins" in each context, beating or hosting, are compared against 10,000 simulations where outcomes are sampled as fair coin flips. 40

1.15 Win probability p_{ij} as a function of score difference $s_i - s_j$ in the Bradley-Terry model as in Eq. (1.54). 41

1.16 Example generative process for the Bradley-Terry model of hierarchies. (a) The depth β is first fixed, then the scores $\boldsymbol{s}$ are sampled from a Gaussian of that width, represented by the vertical distribution with width $\beta = 3$. The differences of these scores then inform who wins each match, where the higher score participant is favored. (b) This same process starting with a depth of $\beta = 1$. This smaller depth leads to smaller differences in scores and so more upset wins, colored in red. 43

1.17 Posterior distribution of the depth β inferred by the general Bradley-Terry in the football victories and football hosting data sets. There is strong evidence of a hierarchical structure among the pattern of victories, while the pattern of hosting does not exclude the possibility that outcomes are fully random, highlighted at $\beta = 0$. 44

1.18 Depths of hierarchies inferred by our model across various applications. A depth of $\beta = 0$ corresponds to outcomes determined by fair coin flips while higher values of β indicate deeper, stricter hierarchies. Full posterior distributions of β for these cases are given in Figure 4.2. 45

2.1 Examples of degree distributions $\boldsymbol{k}$ sampled from the prior distributions of the (a) non degree-corrected and (b) degree-corrected SBMs. These examples contain two groups of 1000 nodes each, group 1 (orange) has average degree 10, and group 2 (blue) has average degree 30. In the non degree-corrected case, the multinomially distributed degrees are concentrated about their group means and the two groups have little overlap in degree. In the degree-corrected case the degree distribution within each group is much broader, leading to greater overlap between the groups. 52

2.2 Inferred group structures of the political books network [1]. The size of each node represents its degree and the vertical position roughly indicates each book's political lean. The node colors represent groups found by each model. The (a) non degree-corrected SBM places high and low degree nodes into separate groups, the (b) degree-corrected version finds groups spanning many degrees, and the (c) general degree-corrected model has a mixed behavior. . . . 54

2.3 Total and within-group degree inequality as assessed by the configuration model and general degree-corrected SBM. 58

2.4 Figure of the log posterior-predictive performance of the degree-corrected models discussed in this work across 50 cross-validation splits of the models relative to the usual non degree-corrected SBM. Data sets are arranged in order of increasing degree inequality G_{in}. 59

2.5 Description lengths of an example network given two potential partitions. The "true" partition into the 8 connected components is compared to a coarsening into only 4 groups. The usual SBM prefers the coarsened partition, indicated by the smaller description length, since the resolution limit prevents it from identifying the small groups. The assortative SBM does not have this limitation and finds the true partition. 63

2.6 Examples of the generative process of the edge count matrix $\boldsymbol{M}$ in the (a) traditional SBM, (b) simple assortative SBM, and (c) planted partition model. In each case the density parameter(s) are first sampled, followed by the weight matrix $\boldsymbol{\omega}$, which generates the observed edge count matrix $\boldsymbol{M}$. Depending on the model, these steps produce final edge counts with and without an assortative preference (high diagonal counts) and to different levels of variation in the entries. 67

2.7 (a) Examples of two types of synthetic networks and their edge count matrices. In the "Symmetric" case all pairs of groups are equally likely to connect while only neighboring groups connect in the "Ring" case. (b) Results of fitting the models presented in this work to synthetic networks of each type as the true number of groups ranges from 10 to 100. Inferred partitions are compared against the truth by comparing the number of groups found and using the normalized mutual information [66]. In both tests the traditional SBM is unable to detect more than ~ 40 groups. In the symmetric case this is caused by the resolution limit while in the ring case it is due to insufficient out-group variation v_{out}. While the planted partition and hybrid assortative models perform well in the symmetric case, they perform very poorly for the ring cases. The general ASBM performs well across all cases considered. 71

2.8 Comparison of the number of groups q_{SBM} found by the traditional SBM with the number q_{ASBM} found by the simple assortative SBM in the networks of Table 2.2. The assortative model consistently identifies more groups than the usual SBM, suggesting that the usual SBM is hampered for these networks by the resolution limit. The point colors are meant only to help distinguish the data sets. 73

2.9 Typical in-group and out-group densities inferred by the general assortative SBM for the data sets listed in Table 2.2. Most of the data sets are assortative, $\rho_{\text{in}} > \rho_{\text{out}}$, while some data sets are mildly disassortative, $\rho_{\text{in}} < \rho_{\text{out}}$. Especially in the assortative networks, the posterior distributions of these parameters are well separated from the non-assortative case $\rho_{\text{in}} = \rho_{\text{out}}$ and so exclude the usual non-assortative SBM. 74

2.10 Variation in the in-group and out-group weights v_{in} and v_{out} as assessed by the general assortative SBM. The corresponding in and out-group homogeneity parameters λ_{in} and λ_{out} are plotted as well. 76

2.11 Figure of the log-posterior-predictive performance of the assortative models discussed in this work across 50 cross-validation splits of the models relative to the traditional (non-assortative) SBM. Data sets are presented from left to right in order of increasing assortativity $\rho_{\text{in}}/\rho_{\text{out}}$. Downward arrows indicate that model performance is below the plotted range. The general and simple ASBM presented in this work consistently outperform alternative models in these tests, particularly for highly assortative data sets. 77

3.1 A contingency table for two labelings (colors) of the nodes of a network, one with three colors and one with two. The entries in the 3 × 2 contingency table $n^{(gc)}$ count the number of nodes that have each pair of label values. The row and column sums $n^{(g)}$ and $n^{(c)}$ then count the number of nodes with each label in the two labelings. Note that, although we illustrate the contingency table with an application to a network, the table itself is a function of the labelings only and is independent of the network structure. 84

3.2 Optimal values of α_g for the transmission of the community sizes $n^{(g)}$, along with the resulting information cost in bits $H_{\text{DM}}(n^{(g)})$ in our new Dirichlet-multinomial encoding scheme and the corresponding cost $H_{\text{flat}}(n^{(g)})$ in the old, flat encoding. Note how vectors with more extreme values benefit from smaller values of α_g, while more uniform vectors favor larger α_g. 91

3.3 Comparison of the information cost of transmitting example contingency tables under the old and new encodings. The top half of the figure shows the new encoding, along with the values for the optimal Dirichlet-multinomial parameter $\alpha_{g|c}$ and the resulting information cost $H_{\text{DM}}(n^{(gc)}|n^{(c)})$. The bottom half shows the old encoding and associated information cost $H_{\text{flat}}(n^{(gc)}|n^{(c)})$. The new encoding is more efficient in every case, and especially so in the case of equal labelings $c = g$. 93

3.4 An example where the three normalized mutual information measures considered here differ substantially. A set of objects (circular dots) is divided into three ground-truth groups, represented by the horizontal stripes of red, green, and blue. Three competing candidate divisions of the same objects are represented by the black boxes denoted A to I. In each, the ground-truth groups are divided into a set of equally sized subgroups. But regardless of how many objects are in each subgroup, the unreduced measure NMI_0 returns a maximal score of 1 for all the candidate divisions, while the reduced measures rightfully return lower scores, except in the case where the sizes of the subgroups diverge. For subgroups of intermediate size, such as the groups of size three in the middle column, the Dirichlet-multinomial measure $\mathrm{NMI}_{\mathrm{DM}}$ of this section can give a substantially different and larger score than the standard ("flat") reduced mutual information $\mathrm{NMI}_{\mathrm{flat}}$. 97

3.5 Comparison of the two normalized reduced mutual information measures considered here for the example system in Fig. 3.4, for various sizes $n_r^{(c)}$ of the subgroups as denoted by the labels. In all cases except $n_r^{(c)} = 1$ and ∞ the flat reduced mutual information $\mathrm{NMI}_{\mathrm{flat}}$ returns a lower value than the Dirichlet-multinomial reduced mutual information $\mathrm{NMI}_{\mathrm{DM}}$ because it overestimates the information cost of the contingency table. 99

3.6 Two candidate labelings of the same set of objects. In this figure, 19 objects are divided among four ground-truth groups, represented by the horizontal stripes of red, green, blue, and magenta, and two candidate labelings c_1 and c_2 are denoted by the boxes labeled A to G. The Dirichlet-multinomial measure of this section favors the left labeling c_1 while the flat reduced mutual information prefers the right one c_2. 100

3.7 Fractional reduction in information cost when transmitting the group sizes $n^{(g)}$ under the Dirichlet-multinomial encoding versus the flat encoding. The horizontal axis is linear from 0 to 0.01 and logarithmic above 0.01. The bar chart is a simple histogram of the relative frequencies, while the curve shows the same results smoothed using a quartic kernel density estimator with the same bin width. 101

3.8 (a) Fractional change in the information cost of transmitting the contingency table $n^{(gc)}$ using the Dirichlet-multinomial encoding compared to the flat encoding. The different curves show the distribution of values for different ranges of similarity between the ground-truth and candidate labelings, as measured by the (Dirichlet-multinomial) normalized mutual information. The horizontal scale is linear between -0.1 and 0.1 and logarithmic outside that range. (b) Fractional change in the mutual information from the improved encoding of the contingency table and group sizes. The horizontal axis is linear across the entire range and the different curves again indicate distributions for different ranges of normalized mutual information. Cases where both mutual informations give a result of 0, for example when $q_c = 1$, have been removed, since they yield a fractional change of 0/0. 103

3.9 Normalization can impact which labeling is preferred by a mutual information measure. For the ground-truth labeling g (top), the standard unnormalized mutual information $I_0(c, g)$ of Eq. (3.39) (Stirling approximated) gives different scores to the two candidate labelings c_A and c_B, the former receiving a higher score than the latter. By definition, the asymmetrically normalized mutual information will always agree with the unnormalized measure, but the symmetrically normalized measure may not, as is the case here: in this example the symmetric measure scores candidate c_B higher. Note that the network itself plays no role here—it is included only as a visual aid, as the mutual information values depend only on the sizes of the shaded regions. 108

3.10 Values of the various normalized mutual information measures between two example partitions of the same set of 27 objects. The left partition c_1 divides the objects into nine groups of three objects each and is a refinement of the right partition c_2, which divides them into only three groups. The arrows indicate the direction of the comparison and the accompanying notations give the values of the mutual information measures. All three asymmetrically normalized measures capture the intuition that the partition c_1 tells us more about c_2 than c_2 tells us about c_1. 112

3.11 Performance of three community detection algorithms: InfoMap (red) and modularity maximization with $\gamma = 1$ (green) and $\gamma = 10$ (blue)—see key on right. The colors in each panel of this figure indicate which of the three is best able to find the known communities in a large set of LFR benchmark networks, according to the six mutual information measures we consider. Mixtures of red, green, and blue denote the proportions of test cases in which each algorithm performs best. Regions in gray indicate parameter values for which no algorithm achieved a positive mutual information score. 113

3.12 Performance of each of the six community detection algorithms considered here in identifying the known communities in a large set of LFR benchmark networks, as quantified by the asymmetrically normalized reduced information measure $I^{(A)}(c; g)$ proposed in this work. Gray areas indicate parameter values for which NMI < 0, so that the conditional encoding is less efficient than the direct encoding. 115

4.1 Score functions $f(s)$. (a) The bold curve represents the standard logistic function $f(s) = 1/(1 + e^{-s})$ used in the Bradley-Terry model. The remaining curves show the function f_α of Eq. (4.8) for increasing values of the luck parameter α. (b) The score function f_β of Eq. (4.9) for different values of the depth of competition β, both greater than 1 (steeper) and less than 1 (shallower). 120

4.2 Posterior distributions of luck and depth. Each cloud represents the posterior distribution $P(\alpha, \beta | \boldsymbol{A})$ of the luck and depth parameters for a single data set, calculated from the Monte Carlo sampled values of α and β using a Gaussian kernel density estimate. The + signs indicate the expected values $\hat{\alpha}, \hat{\beta}$ of the parameters for each data set. 129

4.3 Fitted functions $f_{\alpha\beta}(s)$ for a selection of the data sets. The bold curve in each case corresponds to the expected values $\hat{\alpha}, \hat{\beta}$, while the other surrounding curves are for a selection of values sampled from the posterior distribution, to give an idea of the variation around the average. 129

4.4 Comparative performance of the model of this chapter and a selection of competing models and methods, in the task of predicting the outcome of unobserved matches in a cross-validation experiment. Performance is measured in terms of the log-likelihood (base 2) of the actual outcomes of matches within the fitted model, which is also equal to minus the description length in bits required to transmit the win/loss data given the fitted model. Log-likelihoods are plotted relative to that of the standard Bradley-Terry model with a logistic prior (the horizontal dashed line). Error bars represent upper and lower quartiles over at least 50 random repetitions of the cross-validation procedure in each case. The arrows along the bottom of the plot indicate cases where the log-likelihood is outside the range of the plot. 137

A.1 Schematics of four perspectives on coin flips. (a) Using statistics to infer the probability of landing heads ("H"). (b) Physical interpretation where tails ("T") is more likely as a lower "energy" state than heads. (c) Encoding the sequence of flips as a binary string to quantify the complexity of the sequence in bits. (d) Cross-validation of a model trained on a subset of the data against a withheld testing set. 142

A.2 Example applications of the tools discussed in this section to network science. (a) Statistical inference of community structure of a given network, as in Section 1.2.3. (b) Markov Chain Monte Carlo to explore the posterior distribution (energy landscape) of possible community structures of a network, as in Appendix B.6. (c) Information theoretic measures of the information content and similarity of two possible network community structures, as in Chapter 3. (d) Prediction of unobserved links in a network, as in Figure 2.4. 143

A.3 Distribution of the number of heads n_H observed over $n = 10$ flips of a fair coin and the resulting p-value probabilities of observing at least n_H heads. The region where the p-value is less than 0.05 is plotted in gray, a threshold to reject the null hypothesis of a fair coin. 145

A.4 (a) Posterior distributions of the parameter p after observing 0 coin flips (a.k.a. prior distribution), 10, and 100 coin flips. In each experiment 40% of trials are heads, a ratio marked by the vertical line in each plot. The posterior distributions are proportional to the product of the likelihood (b) and prior (c) over p. As more observations are made the posterior is increasingly informed by the model likelihood. 147

A.5 Schematic of the full generative process of the fair and biased coin models. A parameter value p is first drawn from the appropriate prior $P(p)$, represented by the thicknesses of the top gray arrows to example parameter values $p = 0.4, 0.5$, and 0.6. This choice of parameter then generates the observed data according to the model likelihood $P(\mathrm{H}, ..., \mathrm{T}|p)$, represented by the weights of the colored arrows to the example observations of all tails, a mixture, and all heads. Lastly we compute the model evidence of each observation by summing over all generative paths leading to that outcome. For some data the fair model performs better while for others the biased coin is preferred. 149

A.6 Physical interpretation of coin flips. Heads ("H") have energy $E = 1$ while tails ("T") have energy $E = 0$, so that the overall energy of the configuration $\boldsymbol{s}$ is equal to the number of observed heads, $H(\boldsymbol{s}) = n_H$. 153

A.7 Thermalization of a subsystem (of coin flips) in contact with a thermal bath. In panel (a) all excited states ("H") are contained in the bath, an unusual configuration with overall entropy $S(E) + \tilde{S}(\tilde{E}) = 8.77$. Panel (b) has a more typical, even arrangement of the excited states, reflected in the higher total entropy $S(E) + \tilde{S}(\tilde{E}) = 10.82$. If the subsystem begins in the ground state (a) it will thus likely thermalize to the equilibrium (b). 155

A.8 Example microstates of the subsystem $\boldsymbol{s}$ grouped into macrostates by energy E. Only one microstate has $E = 0$ while five microstates share $E = 1$, meaning that the higher energy macrostate has higher entropy $S(E)$. Although each microstate is equally likely, the higher entropy macrostate is more likely to be observed. 156

A.9 Trajectory of the Metropolis-Hastings MCMC algorithm to thermalize a system of 100 coin flips. The system begins in the ground state where $n_H = 0$. The average value $n_H = 40$ is highlighted. Once the Markov chain has thermalized, the samples are indicative of this average value. 163

A.10 Examples of an (a) balanced code and (b) optimized Huffman code to convert the phrase "ELEVENELVES" into a binary string. The string associated to each letter is denoted by the path from the top of the tree down to the appropriate node. The Huffman code is able to produce a shorter overall message than the balanced code by representing the common letter "E" with a short string "0" despite representing the uncommon letters "N" and "S" with longer strings. . . 165

A.11 The analogous phase transitions of the Ising model and stochastic block model. (a) The Ising model moves from an ordered phase at low temperature T to a disordered, noisy phase at high temperature. These phases are separated by a critical temperature at which the correlation length diverges. Samples from the model on a square lattice are shown. (b) Inference in the stochastic block model is easy when the probability of connection between groups p_{out} is much smaller than within groups p_{in}. If these probabilities are the same inference is impossible. The two regimes are separated by a detectability threshold that corresponds to the phase transition of the Ising model on that network. 178

B.1 Consensus clustering found by the general degree-corrected SBM on the network of $n = 105$ books shown in Figure 2.2c. (a) Entries of the aggregate coincidence matrix $\bar{\boldsymbol{C}}$ taken from the posterior distribution. (b) The coincidence matrix of the found consensus clustering $\boldsymbol{b}^*$, defined by closeness to the aggregate coincidence. 182

B.2 Sampled degree distributions over $n_r = 10000$ nodes of average degree $m_r/n_r = 10$ from the hierarchical prior and general prior for the concentration parameter α_h. Both distributions are compared against the asymptotic distribution Eq. (B.24). 187

B.3 (a) Plot of the description length of a fully assortative network as the number of true groups q varies. (b) True number of planted groups q and the inferred number of groups q^*. Beyond the resolution limit when $q > q_c$ the usual SBM cannot infer more than the critical number of groups. 190

B.4 Figure of inferred group assortativity $\rho_{\text{in}}/\rho_{\text{out}}$ versus the modularity Q of partitions sampled by the simple ASBM. $Q = 0$ and $\rho_{\text{in}}/\rho_{\text{out}} = 1$ are highlighted to delineate between the assortative and disassortative regimes. 193

C.1 (a) The absolute change in the information cost of transmitting the vector of true group sizes $n^{(g)}$ between the standard flat encoding and the optimized Dirichlet-multinomial encoding. In contrast with Fig. 3.7, the information cost $H(\alpha_g) = 4$ bits for transmitting the Dirichlet-multinomial parameter is included in this comparison. (b) The absolute change in the information cost of transmitting the contingency table $n^{(gc)}$ between the standard flat encoding and the Dirichlet-multinomial encoding (including $H(\alpha_{g|c})$). The different curves show the distribution for different levels of similarity between the ground-truth and candidate labelings, as measured by the normalized mutual information. The horizontal scale is linear between -10 and 10 and logarithmic outside that range. In both panels the densities of cases are transformed and smoothed as in Fig. 3.7. 203

C.2 The number of groups inferred by each of the six algorithms in Fig. 3.11 for LFR benchmark networks with $n = 3200$ nodes and a range of values of the mixing parameter μ. The true planted number of groups is shown in black. The InfoMap algorithm (red) generates an accurate number of groups for values of μ up to about 0.5, but beyond this point it erroneously places all nodes in a single group. Standard modularity maximization with resolution parameter $\gamma = 1$ (green) underestimates the number of groups, presumably because of the resolution limit on the detection of small groups, but not as severely as the number is overestimated when $\gamma = 10$ (blue). 206

C.3 Performance of each of the six community detection algorithms considered here, as quantified by the conventional symmetrically normalized non-reduced mutual information $I_0^{(S)}(c; g)$. 208

D.1 Results from the same set of cross-validation tests shown in Fig. 4.4, but quantified using (a) accuracy and (b) log posterior-predictive probability, instead of log-likelihood. All results are measured relative to the Bradley-Terry model with a logistic prior, which is represented as the dashed horizontal line in each panel. Error bars represent upper and lower quartiles, estimated from at least 50 random repetitions of the cross-validation procedure in each case. The maximum likelihood and SpringRank models are not included in the lower comparison, since they are based on point estimates rather than Bayesian methods and hence one cannot calculate a posterior-predictive probability. Arrows indicate results that are off the scale. 216

D.2 Absolute log-likelihood and accuracy values per match in the cross-validation tests of Fig. 4.4. This figure differs from Fig. 4.4 in showing absolute values rather than values relative to the Bradley-Terry model with logistic prior. 224

D.3 The data sets of Fig. 4.2 with dashed lines representing the contours of average entropy per match. Low entropy indicates confidence about the outcome of a match; high entropy indicates unpredictability. 225

D.4 The function $W(\beta_S, s)$ of Eq. (D.21) plotted against s, for various values of β_S as indicated. 227

List of Tables

1.1 Table of performance metrics of the Erdős-Rényi, configuration, and general configuration models on the Facebook friend and College football network examples. The Bayesian evidence $H(\boldsymbol{A})$, minus log posterior-predictive $H(\boldsymbol{A}^{\text{test}}|\boldsymbol{A}^{\text{train}})$, and best-fit parameter $\hat{\alpha}_{\text{MAP}}$ in the general model and corresponding Gini coefficient $\hat{G}_{\text{MAP}}$ are given. The log posterior-predictive represents the median result out of 50 cross-validation splits, as described in Appendix B.8. 25

1.2 Summary of the parameter values of the general configuration model that correspond to the other models it generalizes. The degree homogeneity α and corresponding degree inequality G are inferred in the general configuration model. 28

1.3 Table of the number of communities $\hat{q}$ found by the SBM, the Bayesian evidence $H(\boldsymbol{A})$, and posterior-predictive $H(\boldsymbol{A}^{\text{test}}|\boldsymbol{A}^{\text{train}})$ for Erdős-Rényi and stochastic block models on the Division IA and Big Ten conference football match networks. 37

1.4 Table of best the measured depth $\hat{\beta}_{\text{MAP}}$, Bayesian evidence $H(\boldsymbol{A})$, and posterior-predictive $H(\boldsymbol{A}^{\text{test}}|\boldsymbol{A}^{\text{train}})$ for the fair coin and Bradley-Terry (B-T) models on the football wins and hosting data sets. 43

2.1 Summary of the parameter values of the general degree-corrected SBM that correspond to the models it generalizes. For each model the in-group degree homogeneity α_{in} and corresponding in-group degree inequality G_{in} are given. The general model infers these parameters from the network. 56

2.2 Data sets in order of increasing in-group degree inequality $\hat{G}_{\text{in}}$. The number of nodes n, edges m, and a short description are given for each network. 57

2.3 Summary of the parameter values of the general assortative SBM presented in this work that correspond to the models it generalizes. In the traditional SBM both the in-group and out-group densities must be the same, $\rho_{\text{in}} = \rho_{\text{out}}$, although this overall density is inferred. For the assortative SBMs and planted partition model, each density is separately inferred from the data. These models are then distinguished by their homogeneity parameters λ_{in} and λ_{out}, which may also be expressed as the variation parameters v_{in} and v_{out}. 68

4.1 Data sets in order of increasing depth of competition β. Here n is the number of participants and m is the number of matches/interactions. Further information on the data sets is given in Appendix D.19. 130

4.2 Summary of the parameter values of the joint luck-plus-depth model presented in this paper that correspond to the other models it generalizes. We note that here the Bradley-Terry (B-T) model with the logistic prior is only approximately represented by our model with $\beta \approx 2.56$, as detailed in Appendix D.22. 134

B.1 Data sets of Table 2.2 along with an inter-quartile range of the inferred Gini coefficient of in-group degree inequality G_{in}. These are compared against the values of the index G_h and the concentration parameter α_h that would asymptotically recover the behavior of the hierarchical degree prior as in Eq. (B.31). Most cases exclude this special case in favor of the general degree-corrected model. 188

D.1 Inferred parameter values for the data sets considered in Section 4.4. From left to right: $\hat{\beta}$ is expected depth, β^* is the jointly optimized MAP depth as in Eq. (D.5), $\hat{\beta}_{\text{range}}$ is depth between the best and worst player as in Eq. (D.6), $\hat{\beta}_0$ is restricted depth as inferred in the depth-only ($\alpha = 0$) model, std($\hat{\boldsymbol{s}}_L$) is the standard deviation of the MAP scores within the logistic-prior model, S_{DS} is the steepness measure of de Vries *et al.* [41], S_{Elo} is the steepness measure of Neumann and Fischer [96], $\hat{\beta}_S$ is the maximum likelihood estimate of the parameter β_S in the SpringRank model [40], $\hat{\alpha}$ is the expected luck, and α^* is the jointly optimized MAP estimate of the luck. 222

List of Appendices

Appendix A. Supplementary Background Material . . . 141

A.1 Statistical inference . . . 141
A.2 Statistical physics . . . 151
A.3 Information theory . . . 163
A.4 Prediction and validation . . . 172
A.5 Equivalence of SBM inference and the Ising model . . . 176

Appendix B. Supplementary Material for Chapter 2 . . . 179

B.6 Stochastic block model Markov Chain Monte Carlo . . . 179
B.7 Consensus clustering . . . 181
B.8 General posterior-predictive . . . 182
B.9 Hierarchical degree prior . . . 184
B.10 Resolution limit . . . 187
B.11 Synthetic tests . . . 190
B.12 Comparison to modularity . . . 192

Appendix C. Supplementary Material for Chapter 3 . . . 195

C.13 Upper bound on the normalized mutual information . . . 195
C.14 Clustering and permutation invariance . . . 198
C.15 Choosing and transmitting the Dirichlet-multinomial parameter . . . 203
C.16 Benchmarking . . . 205
C.17 Bias towards large numbers of groups in the traditional mutual information 209
C.18 Adjusted mutual information . . . 211

Appendix D. Supplementary Material for Chapter 4 . . . 213

D.19 Data sets . . . 213
D.20 Other measures of model performance . . . 214
D.21 Point estimates of parameters . . . 217
D.22 Other measures of depth . . . 218
D.23 Depth as predictability . . . 221
D.24 SpringRank . . . 225

Abstract

Many systems across science can be meaningfully represented as networks of simple interactions. Within networks of metabolic pathways, transportation links, neuronal connections, outcomes of sports matches, and social ties, the patterns and directions of data often exhibit collective behavior. In this thesis we focus on characterizing two common structural features of these networks: group structure and hierarchy.

In real-world networks we often find communities — groups of nodes which interact with each other more frequently than with nodes outside their group. Common examples include friend groups, functional neuronal groups, or ecological niches. Despite their ubiquity, network data does not often come to us already labeled with this group structure; it must be algorithmically inferred from the network alone. In this work, we describe a number of refinements of existing community detection models which allow us to: discover more and smaller groups, directly measure the in-group preference within the system, and measure the variation in connections within groups.

When the ground truth community structure of a network is available, the outputs of these community detection algorithms are often compared against that truth in an information theoretic manner. We discuss improvements to this framework that address a bias towards algorithms which find an excessive number of communities and better quantify relevant information costs. We further demonstrate how the conclusions drawn depend on the form of this measure in extensive tests on synthetic networks.

Similarly, when we observe directed relationships such as dominance interactions among animals or humans, the directions of faculty hiring among universities, or wins and losses in games and sports, hierarchies routinely emerge. In this work we draw an analogy between fermion energy and competitor rank in these hierarchies to define a notion of the collective "temperature" of a hierarchy which we may then measure. This parameter then also indicates the strictness of the hierarchy, or equivalently the number of distinct levels of play. We find a good deal of variation in this measured temperature: sports rankings tend to be hot and unpredictable, animal hierarchies are cold and rigid, while human social hierarchies are lukewarm — somewhere in the middle.

Taken together, our work has not only enabled new insights about longstanding prob-

lems of network science but also offered a path to understand and measure their nature.

Chapter 1

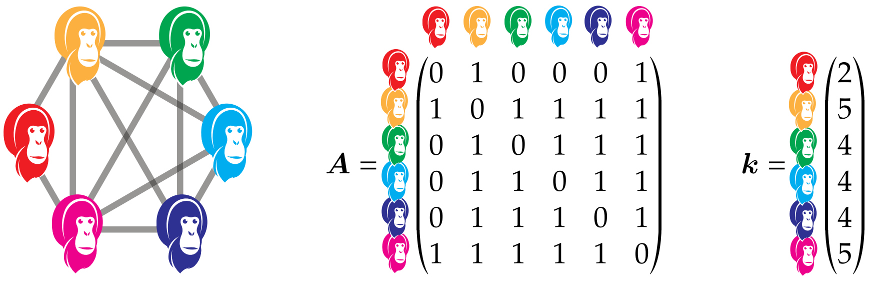
Introduction

1.1 Networks and complex systems

Networks are often used to represent complex systems composed of many parts, such as individual people, animals, or atoms. Their study focuses on the nature and structure of interactions between these components. Just as a universe of only non-interacting particles would be a lifeless soup, a collection of people who never interact is boring and unrealistic to study. The network of interactions between the many pieces makes the system “complex.”

1.1.1 Representing network data

In a network (or "graph") these components are abstractly represented as *nodes* ("vertices") and the interactions are represented as *edges*, lines that connect interacting nodes with each other. This simple framework is flexible enough to span a rich variety of applications and data under various interpretations of the nodes and edges.



(a) (b) (c)

Figure 1.1: Network of interactions observed by Setia *et al.* [120] among $n = 6$ orangutans represented (a) as a graph and (b) as an adjacency matrix $\boldsymbol{A}$. (c) The node degrees $\boldsymbol{k}$, the number of interactions of each orangutan.

An example network is given in Figure 1.1a. There the colored nodes represent 6 Indonesian flanged male orangutans who are connected by an edge if a vocalization between them was observed in a study by Setia *et al.* [120]. As reflected in the graph, 5 of the 6 orangutans form a *clique* of nodes that each interact with each other while the remaining ape represented in red interfaced with only the orange and magenta orangutans.

Such a network on n nodes is often alternatively represented by a $n \times n$ *adjacency matrix* $\boldsymbol{A}$, whose entry A_{ij} for $i, j = 1, \ldots, n$ is equal to 1 if nodes i and j interact and 0 otherwise. Figure 1.1b shows the orangutan network in this format with the rows and columns labeled by the orangutan they reference. This matrix and its properties naturally convey information about the network. For example the vector $\boldsymbol{k}$ of row (or column) sums of the adjacency matrix

$$k_i = \sum_{j=1}^{n} A_{ij}, \qquad i = 1, \ldots, n, \tag{1.1}$$

gives the *degree* of each node, the total number of edges connected to it.

These degrees of our apes are given in Figure 1.1c. The red orangutan is the only node with degree 2, while the orange and magenta orangutans have degree 5 as they both connect to every other orangutan. This simple measure of counting connections captures a sense of *centrality* within the network as certain orangutans appear more disposed to interact than others. This sort of variation in the degrees is typical of real networks, particularly within large data sets. We further discuss how the distribution of the degrees can be modeled and interpreted in Section 1.2.2.

The edges in Figure 1.1 are bidirectional since they symmetrically represent whether any interaction occurred between a pair of orangutans. The network is therefore an *undirected* graph and has a symmetric adjacency matrix. Although such undirected graphs are very common, and the focus of much of this thesis, many systems possess asymmetric interactions. In fact, in their observations Setia *et al.* not only recorded whether a pair interacted but also which orangutan was dominant over the other from their pattern of vocalizations. We can add this layer of information to the network by converting it to a *directed* graph, where each edge is now represented by an arrow that points from the dominant "winner" of the interaction to the submissive "loser."

The original network of Fig. 1.1 is also a *simple* graph as interactions are binary – two nodes either do or do not connect and no node connects to itself. However some pairs of orangutans interacted more than once over the course of the study, a feature we can represent with a *multigraph* where nodes can connect with more than one edge and *self-*

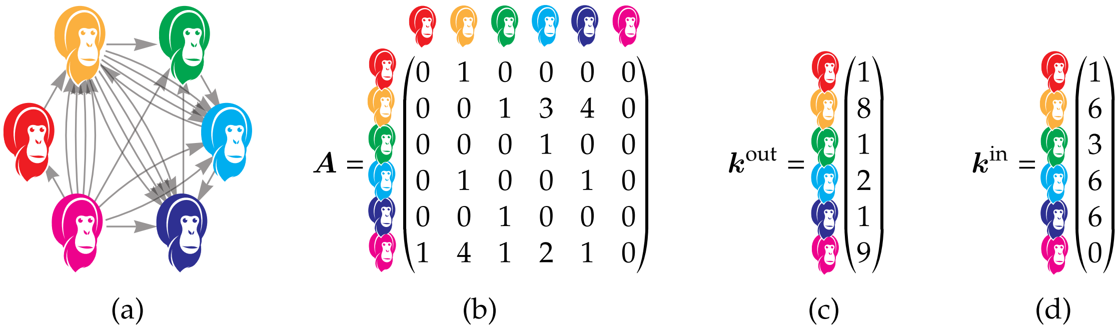


Figure 1.2: Directed multigraph of dominance interactions observed among $n = 6$ orangutans [120] represented (a) as a graph and (b) as an adjacency matrix $\boldsymbol{A}$. (c) The out-degrees $\boldsymbol{k}^{\text{out}}$ and (d) the in-degrees $\boldsymbol{k}^{\text{in}}$ are also given, respectively counting the number of times each orangutan exhibits a dominant or submissive interaction.

edges can connect a node to itself. Although self-edges are not present in the orangutan context, such self-interactions arise in other settings. In a food web of predation among species, for example, cannibalism within a species is represented with a self-edge.

By incorporating these generalizations, the network of Figure 1.2a represents the full detail of the data set of Setia *et al.* [120]. Where the original network of Figure 1.1 represents *if* a pair of orangutans interact, this new network gives fuller context as to the *nature* of the interactions. The adjacency matrix in Figure 1.2b is no longer a symmetric binary matrix as A_{ij} now represents the number of directed edges from node i to node j. This asymmetry generates two distinct notions of degree. Figure 1.2c contains the row sums of the matrix, the *out-degrees* $\boldsymbol{k}^{\text{out}}$, that indicate the number of edges pointing away from a given node, in this case the number of times each orangutan had a dominant interaction. In Figure 1.2d the column sums of the matrix give the *in-degrees* $\boldsymbol{k}^{\text{in}}$, the count of edges pointing towards each node, the number of times each orangutan had a submissive interaction. The directions of the data and resulting degrees give further insight to the social world of the orangutans. For example the magenta orangutan has $k_i^{\text{out}} = 9$, $k_i^{\text{in}} = 0$ and so was dominant in each of his 9 interactions, suggesting that he holds a prime position in the orangutans' social hierarchy.

Further adornments of networks are often made to represent further detail. In a trade network each edge could carry a continuous weight that reports trade volume between two countries, an example of a *weighted graph* represented by a continuous adjacency matrix [45]. Edges can also be differentiated by their type, for example friendships in a social network may be associated to different contexts like work, hobbies, or social media forming a *multilayer* network of connections between people [23, 73].

Higher-order interactions between triplets and larger groups of nodes may be also be represented either as hypergraphs or bipartite networks, although they are beyond our present scope. In this thesis we focus on interactions between only pairs of nodes, so-called *dyadic interactions*. In fact most of this thesis focuses only on the nature of the simple, undirected networks exemplified by Figure 1.1. Only in Section 4 do we consider directions of edges and the patterns they reveal.

Although adding increasingly descriptive context and metadata to networks helps to paint a fuller picture of any particular application, by stripping systems down to a simple pattern of interactions it is possible to treat them within a unified language of network science and understand similarities and differences between them. In this thesis we focus on two common structures found in networks across contexts, group and hierarchical structure.

1.1.2 Group structure

The nodes that make up a network often possess distinct group identities. Animals are distinguished by species, atoms differ by atomic number, and students belong to various clubs and social groups. These group labels may then influence the interactions among the nodes as, for example, students are often friends with others in the same school club. Figure 1.3 gives four examples of real networks where the color of each node indicates which group it belongs to. In each system these group affiliations guide the pattern of connections as nodes of the same group share the same structural role.

Figure 1.3a contains a network of 616 American football matches played among 115 Division IA colleges in the 2000 regular season [57]. Each year these football teams group into conferences that agree to play a certain number of matches, often 8, against other teams in the same conference. While matches out-of-conference are permitted, they occur more rarely as they are organized on an ad hoc basis from year to year. The conference system thus gives teams an *assortative* preference to play teams of the same group. Over the course of the season the network of matches then carries a clear signature of the conferences that inform them.

Similarly assortative structure is found in Figure 1.3b, a network of 105 political books frequently co-purchased on Amazon.com near the 2004 U.S. presidential election. Here the books are colored by the political lean of their content, characterized by V. Krebs in unpublished work. Books of the same political bent are typically purchased together, either both liberal or both conservative. Rarely does someone order both a conservative and liberal book in the data set. In this case the observed assortative structure is not imposed by a collective agreement like the football conference system but rather emerges

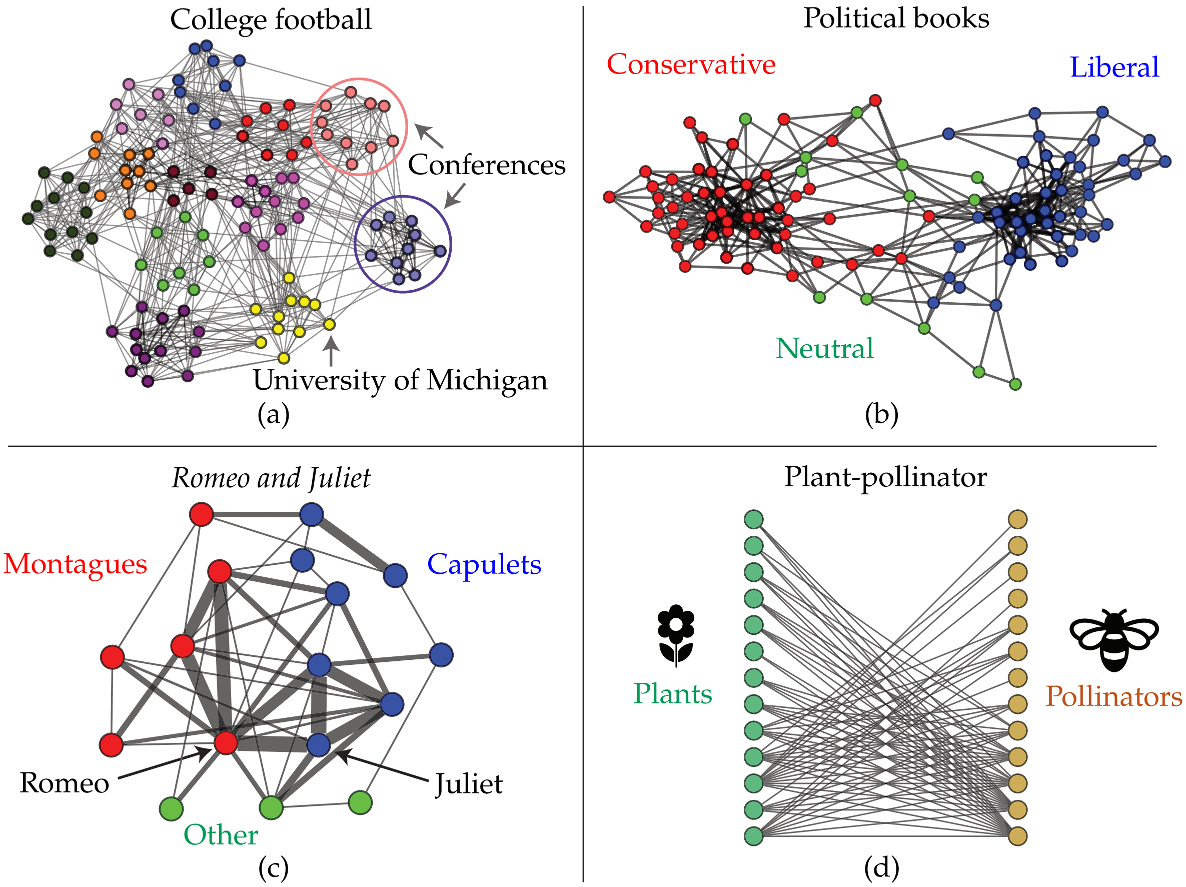


Figure 1.3: Four examples of community structure in networks. (a) American football matches among Division IA colleges in the 2000 regular season [57]. Most matches occur within the labeled "conferences" that the teams belong to. (b) Network of political books frequently co-purchased on Amazon.com near the 2004 U.S. presidential election labeled by political lean. (c) Network of characters in Shakespeare's *Romeo and Juliet* where edge thickness indicates frequency of interaction. The house each character belongs to in the work, Montague or Capulet, is highlighted. (d) Bipartite network of plant-pollinator interactions [16]

.

from individual decisions as an effect of systemic political polarization.

In networks of social ties, people and animals often organize into tightly knit *communities* that prefer to interact amongst themselves. Figure 1.3c contains a (fictional) example of such a network among characters in Shakespeare's *Romeo and Juliet*. Character interactions, defined as subsequent appearances in the same scene, form the edges of this multigraph, where the edge thickness reflects the frequency of the interaction. In the play most characters belong to one of two groups, either Romeo's House of Montague or Juliet's House of Capulet. Interactions between characters are indeed mostly contained within each house. The narrative, however, revolves around the titular exception to this pattern. Later in this thesis we will observe similar patterns of communities across social networks and quantify the extent to which their group boundaries are crossed.

Not all group structures are assortative in this manner. In fact many network data sets are *bipartite*, meaning they consist of two groups of nodes that never interact within their own group, only with the opposite group: a fully *disassortative* structure. Figure 1.3d gives an example of a bipartite plant-pollinator network. In it 13 species of Brazilian oil-flowers are connected to which of 13 species of pollinators visited them over the course of a study by Berezza *et al.* [16]. In this setting the nodes are naturally sorted by their role as either a plant or a pollinator. Being a bipartite network, all interactions occur only between a plant and a pollinator, never directly between two plants or two pollinators.

In these examples and beyond networks possess a variety of group structures, distinguished both in the number of groups present and how those groups inform the pattern of interactions. In the four examples of Figure 1.3, group identities can be gleaned from context outside the network of interactions. However in many contexts only the pattern of connections is known, although nodes may still meaningfully belong to groups that influence that structure.

In this setting a key task of network science is *community detection*, to analyze a network and identify if group structure exists, identify the groups, and characterize their structural relationships. A variety of approaches have been developed for this task, including novel methods discussed in Section 2 of this thesis. The groups inferred from the network can then be interpreted in their own right or compared to known group labels. In Section 3 we will discuss and refine how the difference between these inferred and true groups can be quantified.

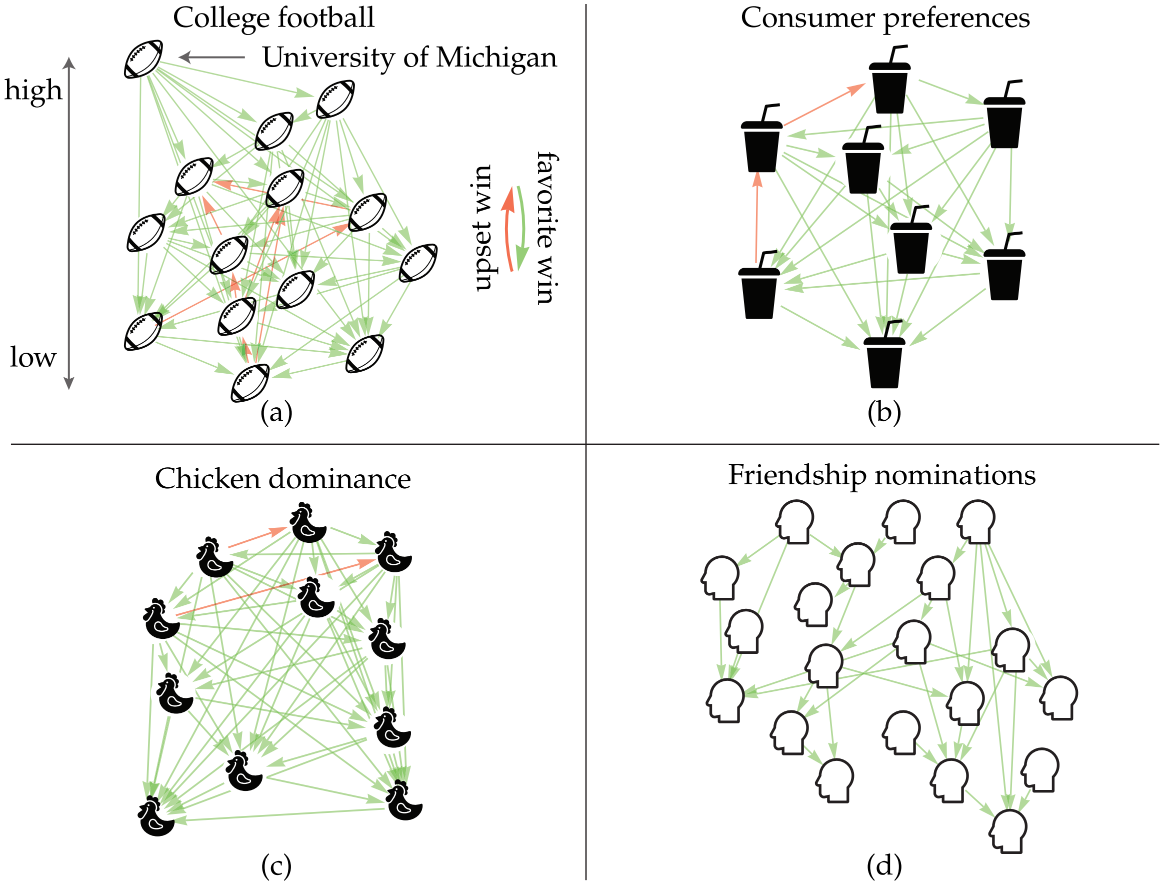


Figure 1.4: Four examples of hierarchy structure in networks. (a) American football matches played in the 2022 Big Ten conference football season [2]. The hierarchical structure is represented vertically, with better teams located higher. Arrows run from the winner to the loser of a match, colored green if the higher ranked team wins, red if the lower ranked team wins. (b) Consumer preferences between pairs of sodas [46]. (c) Observed dominance interactions (pecks) among 11 chickens [87]. (d) Unreciprocated friendship nominations among 20 high school students [128].

1.1.3 Hierarchy structure

As in our orangutan example, network interactions are often directed. Edges in a directed graph variously run from a source to a target, a winner to a loser, a leader to a follower. Across a system these asymmetries often follow a coherent direction within a hierarchy of the nodes. In a sports context this may be a hierarchy of skill where a more skilled player is likely to prevail over a less skilled opponent. In social settings "dominance interactions" tend to be won by the higher status individual. In Figure 1.4, four examples of such directed networks are given. In each case the hierarchy of the nodes is plotted along the vertical axis.

Figure 1.4a contains football matches played within the 2022 Big Ten football conference [2]. Each arrow points from the winner to the loser of a match, information that was suppressed in the undirected Figure 1.3a. The University of Michigan was undefeated in this (carefully chosen) season as indicated by all arrows pointing away from its node and by its position at the top of the football hierarchy. In this case placement on the hierarchy is meant to reflect strength in the game of football, and we would expect that better teams win more often than lower ranked ones. Particularly we may assume that the outcome of any given match is driven by the difference in skill between the participating teams. Although the teams and fans are generally aware of these relative strengths, there is not a "true" hierarchy to refer to like the conference group structure. The positions in the hierarchy must instead be deduced from the observed matches.

There are many ways to determine a hierarchy from the match outcomes. Collegiate football formalizes this inference through an annual poll of team coaches to determine the highest ranked teams who may then advance to the post-season playoff bracket. In Figure 1.4a, the vertical hierarchy is instead arranged in order to maximize the number of times the higher ranked team wins (the 57 green arrows) and minimize the upset victories (the 7 red arrows). The other networks of Figure 1.4 are likewise arranged to minimize such *violations* of their hierarchies. We discuss other methods for arriving at such a ranking in Section 1.2.4.

The presence of upsets means that the data does not fully respect the hierarchy. One might hope that there is some other ordering of the teams that is fully coherent, lacking any such upsets, but the intransitivity of the results prevents this. In the football example Minnesota beat Nebraska, Nebraska beat Iowa, and Iowa beat Minnesota in a rock-paper-scissors arrangement. One of these three matches would therefore be recorded as an "upset" under any ordering of the teams. In fact any ordering of this season will have at least 7 upsets as in the ordering plotted in the figure. This tension between the hierarchy and the realized results is analogous to how not all edges lie within the groups of Figure 1.3. The degree to which the outcomes conform to a hierarchy gives insight to its strictness.

Many of the models used to describe hierarchies were initially developed and are often used in the world of sports, yet similar concepts can be applied to understand status and quality in other contexts. Figure 1.4b shows a hierarchy of 8 different flavored sodas. In this network wins indicate if most assessors surveyed by Duineveld *et al.* prefer one soda over another in a *paired comparison* study [46]. Again no total ordering of the sodas is possible as the revealed preferences are not strictly transitive, although there are certain sodas that generally fare better in these comparisons. Such surveys and inferences are often used to make sense of A/B testing of various types of products and establish the aggregate consensus of quality and preferences. These methods are similarly used in reinforcement learning to assess and tune large language model outputs based upon human preferences [29].

Turning to animals, the social hierarchy of a flock of 11 Brown Leghorn chickens is plotted in Figure 1.4c [87]. Here the arrows represent when one chicken pecks another but is not pecked in return, taken as a sign of dominance. The chickens organize into a pecking order where the higher chicken tends to peck the lower chicken on the ladder. This animal behavior research is in fact the source of the phrase "pecking order" colloquially used to refer to all sorts of hierarchies. In this flock of 11, the chicken at the bottom of the pecking order is pecked by all above it while there is more competition and ambiguity at the top.

Figure 1.4d shows a similar social hierarchy, now of surveyed friendship nominations among 20 students of a small U.S. high school [128]. Although we may think of friendship as a two-way street, this reciprocity is not always borne out in survey data. Often student A names student B as their friend but student B does not name student A back. We can interpret this as a "win" for student B in the social hierarchy, and that they are likely of higher social status than student A. In the school of this figure a clear hierarchy emerges where higher status students consistently dominate the lower ranked students in this manner.

Each of the directed network representations in Figure 1.3 foregoes further context unique to each setting. The football match network, for example, neglects idiosyncratic details like player injuries or home-field advantages that can influence the outcome of any given match. Such events, however, do not have a clear analog within, say, the pecking order of chickens. By reducing the systems down to a directed pattern of "wins" and "losses," we can directly compare them to each other. In Chapter 4 of this thesis we discuss how this framework enables us to observe how the nature and strength of these hierarchies differ across settings.

1.2 Random graph models

This thesis leverages tools from many fields, including statistics, physics, computer science, and information theory to understand networks. Appendix A offers background, motivation, and derivations of many of the key concepts that we employ. The reviews of each of these areas are meant to be concise and self-contained. Readers familiar with these subjects may comfortably skip this background.

In this section, we apply these interdisciplinary techniques to model and measure the group and hierarchy structures exemplified by the networks of Section 1.1. Throughout this analysis we apply these models to analyze selected examples of real networks, demonstrating how they can illuminate various aspects of their network structure.

1.2.1 Erdős-Rényi graph model

We begin our exploration with the Erdős-Rényi (or "flat") random graph model. This simple model serves as the basis of the more detailed structural models we later consider. Although first studied by Solomonoff and Rapoport [122], the model is most closely associated with the work of Erdős and Rényi [50], after whom it is named. In their foundational work, both collaborations showed that despite its simplicity the Erdős-Rényi model captures key properties observed in real networks.

The Erdős-Rényi model is often defined over *simple graphs*, characterized by binary adjacency matrix entries $A_{ij} \in \{0,1\}$. In the *canonical* model, denoted as $G(n,p)$, there is a fixed probability $p \in [0,1]$ that an edge exists between any of the $\binom{n}{2}$ possible pairs of distinct nodes. Statistically, each entry follows an independent Bernoulli distribution, akin to flipping $\binom{n}{2}$ independent biased coins, where the probability p of heads represents the existence of an edge. Collecting these chances, the model *likelihood* of generating a full network $\boldsymbol{A}$ given a choice of parameter p is

$$P(\boldsymbol{A}|p) = \prod_{i<j} p^{A_{ij}}(1-p)^{1-A_{ij}} = p^m(1-p)^{\binom{n}{2}-m}. \tag{1.2}$$

When applying this model, we infer the latent parameter p from our observation $\boldsymbol{A}$. A common *frequentist* approach is to identify the parameter value that maximizes the model likelihood of the data. This optimum is known as the *maximum-likelihood* (ML) estimate

$$\hat{p}_{\mathrm{ML}} = \frac{m}{\binom{n}{2}} \approx \frac{2m}{n^2}. \tag{1.3}$$

This value, equal to the empirical *density* of the network, provides insight into the network's formation and can be extrapolated to make predictions about future observations.

In certain contexts, however, this maximum-likelihood estimate can be misleading. Especially when we observe a small amount of data, there is an inherent ambiguity about the "true" parameter value. Many possible values of p could have generated the network $\boldsymbol{A}$, yet we can only draw conclusions from the single observation. Reporting only the point estimate $\hat{p}_{\mathrm{ML}}$ may not represent the full range of possible parameter values.

To incorporate this uncertainty, we adopt a *Bayesian* approach throughout this thesis. Instead of focusing on a single point estimate, this framework represents our belief about likely parameter values as a distribution over p. To implement this, we first define a *prior* distribution that contains our initial assumptions of reasonable parameter values before observing any data. For our purposes, we use a uniform, *maximum-entropy* prior over the interval, $P(p) = 1$. This choice is as agnostic as possible, treating each potential value of p as equally plausible before considering the data.

Using Bayes' law, we then calculate the *posterior* distribution of likely parameter values given our observed network

$$\begin{aligned} P(p|\boldsymbol{A}) &= \frac{P(\boldsymbol{A}|p)P(p)}{P(\boldsymbol{A})} \\ &= \left[\binom{n}{2} + 1\right] \binom{\binom{n}{2}}{m} p^m (1-p)^{\binom{n}{2}-m}. \end{aligned} \tag{1.4}$$

Here, the distribution is normalized by the *model evidence*

$$\begin{aligned} P(\boldsymbol{A}) &= \int P(\boldsymbol{A}|p)P(p)dp \\ &= \frac{1}{\binom{n}{2}+1} \binom{\binom{n}{2}}{m}^{-1}, \end{aligned} \tag{1.5}$$

an important quantity we will return to later this section. In this model, the posterior $P(p|\boldsymbol{A})$ is maximized at the same value as the likelihood, $\hat{p}_{\text{MAP}} = \hat{p}_{\text{ML}}$, but the width of the posterior distribution reflects the certainty of our inference.

For instance, we can apply this Bayesian model to a real social network of 2218 friendships among 324 Facebook profiles [17] illustrated in Figure 1.5a. Figure 1.5b displays the posterior distribution $P(p|\boldsymbol{A})$ from Eq. (1.4) for this network, showing likely values of the connection probability p. Since this is a large data set, the posterior is tightly peaked around its maximum, $\hat{p}_{\text{MAP}} \approx 0.042$, although nearby values of the parameter are not entirely excluded. As more data is gathered, this posterior naturally narrows as we become increasingly confident in our inference — a concept further demonstrated in Figure A.4.

The average degree of this network is $c = \langle k \rangle \approx 13.7$, meaning each profile, on average, is friends with just a small fraction of the $n = 324$ potential nodes. This behavior is typical of large networks, where the number of connections of each node c tends to stay constant as the number of nodes n grows. For example, an individual is likely to maintain roughly the same number of friendships whether they live in a town of ten thousand of a city of ten million. In most contexts it is neither realistic nor sustainable for the mean degree to grow proportionally with the size of the network indefinitely.

In large networks this bound also implies that the probability p of any specific pair of

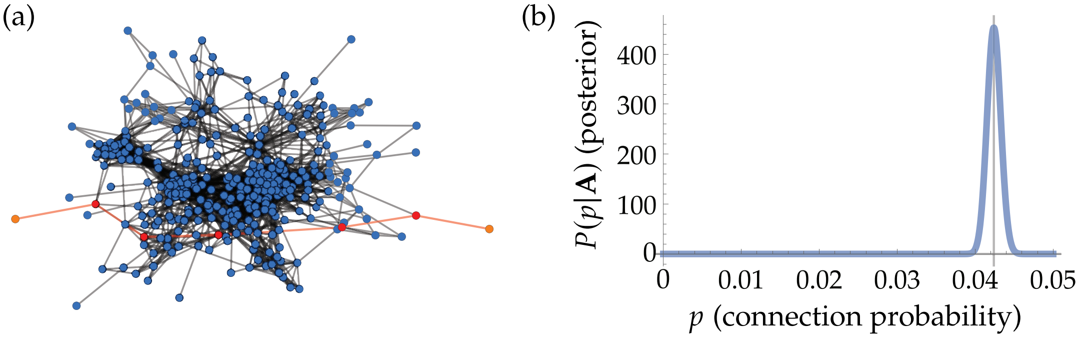


Figure 1.5: (a) A social network of friendships among Facebook profiles [17]. A path of 6 edges, highlighted in red, connecting the leftmost and rightmost nodes. (b) Posterior distribution $P(p|\boldsymbol{A})$ of the connection probability p for this network in the Erdős-Rényi model. The MAP estimate $\hat{p}_{\mathrm{MAP}} \approx 0.042$ is marked by the vertical line.

nodes being connected must be very small. In the Erdős-Rényi model, the average degree is

$$c = p(n-1) \tag{1.6}$$

since each node is connected by an average of p edges to each of the $n-1$ other nodes. Therefore in the *constant degree* limit $c \sim \mathrm{O}(1)$, the probability p decreases as $p \sim \mathrm{O}(1/n)$. As this connection probability approaches 0, we refer to this as a *sparse* limit, contrasting with *dense* graphs that retain a markedly positive density p even in a large system.

Within this realistic regime, the Erdős-Rényi random graph helps to explain the global connectedness observed in real networks. In the sparse Facebook network depicted in Fig. 1.5a, even though the probability $p \approx 0.04$ of a direct connection is small, a path of friendships connecting any two profiles can still be found. The figure highlights a path spanning six friendships between the leftmost and rightmost nodes. Even in very large networks, where each node connects to a vanishingly small fraction of the others, it is often possible to trace a (surprisingly short) path between any two nodes. This phenomenon is famously echoed in the "six degrees of separation" notion popularized by Stanley Milgram's small world experiment, which suggests that any two people can be linked by six social connections [90, 134]. Although this is not strictly true empirically (e.g. some pairs in our Facebook network require seven connections), the spirit tends to hold in social data.

In network language, a path of links between two people indicates that they belong to the same *connected component* of nodes. In a general graph, not all nodes need to be connected to each other in this manner. For example, some individuals live in communities

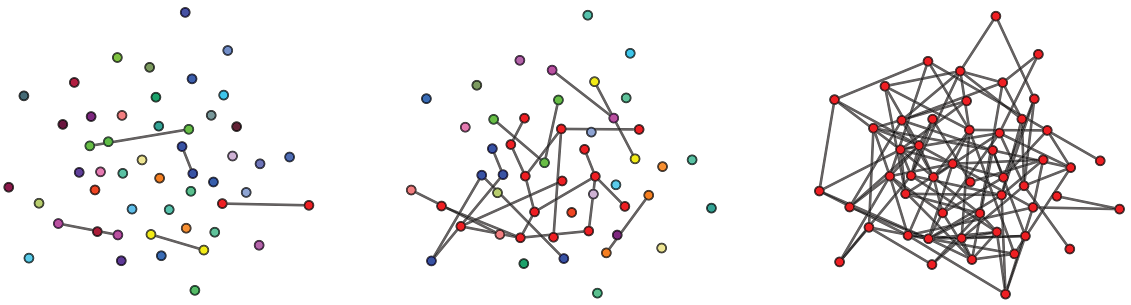


Figure 1.6: Samples of an Erdős-Rényi random graph on $n = 50$ vertices at edge probabilities $p = 0.004, 0.02, 0.1$ corresponding to average degrees $c = 0.2, 1, 5$. Nodes are colored by the connected component they belong to. The middle panel shows the emergence of the giant component at the transition point $c = 1$.

that are entirely isolated from the rest of the world. In such cases, the nodes of a network may split into multiple connected components. Typically, however, one of these components will encompass the majority of the nodes, as most individuals can indeed connect to most others. This prominent sub-network is known as the *giant connected component* and is a feature of most realistic networks.

The common emergence of a giant component can be understood through the Erdős-Rényi random graph model. As illustrated in Figure 1.6, when a network has very few edges they are each isolated, and the connected components are single nodes or edges (represented by different colors in the figure). As additional edges are added, these smaller connected components begin to coalesce into a single giant component which ultimately connects nearly every node in the network.

Between these extremes, the Erdős-Rényi random graph undergoes a *phase transition* at average degree $c = 1$. When $c < 1$, the connected components tend to be small and isolated, of size $O(1)$ in the large n limit. As c surpasses 1, a giant component suddenly emerges, forming a connected component that grows extensively as $O(n)$ with the size of the network and so contains a persistent fraction of all nodes [122].

Once the giant component appears it is also quite easy to traverse. The maximum number of steps needed to connect any pair of nodes, known as the *diameter*, grows slowly as $\mathrm{O}(\log n)$, reflecting the small-world phenomenon [50]. Despite its simplicity, the Erdős-Rényi model effectively captures and explains the connectivity properties observed in real complex systems.

Although the Erdős-Rényi random graph reflects this common global property, we often seek a more comprehensive measure of how well the model captures the full details of the network. To evaluate this, we use the model evidence (or "marginal likelihood") $P(\boldsymbol{A}) = \int P(\boldsymbol{A}|p)P(p)dp$ of Eq. (1.5). This represents the total probability that the random graph could generate the network $\boldsymbol{A}$, integrated over all possible values of the unknown parameter p. By integrating out this uncertainty, the model can be interpreted as a *nonparametric* distribution $P(\boldsymbol{A})$ over all possible networks we could observe.

Model evidence serves as a natural criterion for model assessment. When comparing two models, we prefer the one with the higher evidence as it is more likely to reproduce our observation. Alternatively, as discussed in Appendix A.3, the distribution $P(\boldsymbol{A})$ corresponds to an encoding of the network, compressing its information content into a binary string of length $H(\boldsymbol{A}) = -\log_2 P(\boldsymbol{A})$[1].

The model with the higher evidence $P(\boldsymbol{A})$ will thus have a shorter *description length* $H(\boldsymbol{A})$, and yield a more efficient, parsimonious compression. When modeling, we can therefore equivalently pursue models with higher evidences or shorter description lengths of our observations. For the Erdős-Rényi model, the description length is

$$\begin{aligned} H(\boldsymbol{A}) &= -\log P(\boldsymbol{A}) \\ &= \log\left[\binom{n}{2}+1\right] + \log\binom{\binom{n}{2}}{m}, \end{aligned} \tag{1.7}$$

resulting in a $H(\boldsymbol{A}) \approx 13249.6$ bit compression of the network of Facebook friends. In later sections, we will observe how more realistic network models can more efficiently compress real networks as they incorporate aspects of their structure beyond the overall density.

The form of the model evidence in Eq. (1.5) also suggests an alternative *microcanonical* formulation of the model[2]. In this telling, known as $G(n, m)$, the network is uniformly distributed among all simple graphs with n nodes and m edges. Given the number of ways that the m indistinguishable edges can be arranged among the $\binom{n}{2}$ potential node pairs, the probability of any specific configuration is

$$P(\boldsymbol{A}|m) = \binom{\binom{n}{2}}{m}^{-1}. \tag{1.8}$$

Here, the global number of edges m replaces the local probability p as the key parameter.

[1] Throughout this thesis, information will be reported in bits (base 2), though the choice of logarithm base only changes results by a constant factor.

[2] The canonical and microcanonical designations of these models come from the corresponding ensembles over configuration space in equilibrium statistical physics. See Appendix A.2 for more details.

Since m can be any integer from 0 to a maximum of $\binom{n}{2}$ when all possible pairs are connected, we can again adopt a uniform prior over the possibilities

$$P(m) = \frac{1}{\binom{n}{2} + 1}. \tag{1.9}$$

Together these distributions give the same model evidence as the canonical model,

$$P(\boldsymbol{A}) = P(\boldsymbol{A}|m)P(m) = \frac{1}{\binom{n}{2} + 1} \binom{\binom{n}{2}}{m}^{-1}. \tag{1.10}$$

Thus, we can describe the formation of any network through two ultimately equivalent Erdős-Rényi stories. Canonically, a connection probability p is first randomly chosen, then each edge independently appears with this probability. Microcanonically, the total number of edges m is first randomly set, then the m edges are collectively shuffled among their possible positions.

Throughout this thesis, we describe many other network models using these equivalent formulations, each offering a unique perspective on the network formation process. In the canonical view, a network's structure emerges from many small, independent decisions. In contrast, the microcanonical framework describes network formation as governed by one or more global constraints.

Depending on the context, either formulation may be more appropriate. Certain networks are designed under strict microcanonical constraints. For example, during a sports season, there is typically a fixed total number of games played, but the specific matchups among teams are shuffled. Microcanonical models are also commonly used as *null hypotheses*, allowing networks to be compared against variants that share certain observed properties but are otherwise randomized. Outside these applications, this thesis primarily presents models from a canonical perspective. This picture aligns with the view of complex systems as decentralized and self-organizing, where emergent structures result from local interactions rather than predefined global constraints.

Beyond the simple graphs we have discussed thus far, we will also consider a variant of the Erdős-Rényi model defined over multigraphs in this thesis. This adaptation accommodates scenarios where pairs of nodes are joined by multiple edges or where nodes connect to themselves. This version of the model is not only more flexible than the simple graph model but also more analytically tractable, a useful base for the more complex network models we will build.

In this generalized model, the off-diagonal entries of the adjacency matrix are each independently sampled from a *Poisson distribution* with expectation ρ, rather than a Bernoulli distribution with expectation p. The Poisson distribution admits all non-negative integers $A_{ij} \in \{0, 1, ...\}$ rather than only binary values. Here, the parameter $\rho \geq 0$ sets the *density* of the graph as the typical number of edges between each pair of nodes.

Self-edges, now present in the model, lead to non-zero diagonal elements A_{ii} equal to twice the number of edges connecting node i to itself. This convention ensures that the row sums $\boldsymbol{k}$ of the adjacency matrix equal the vertex degrees. We model the number of self-edges $A_{ii}/2$ as a Poisson distribution with expectation $\rho/2$, ensuring the diagonal elements A_{ii} are even while all matrix entries share the expectation ρ. Collecting these assumptions, the multigraph likelihood is then

$$\begin{aligned} P(\boldsymbol{A}|\rho) &= \prod_{i<j} \frac{\rho^{A_{ij}} e^{-\rho}}{A_{ij}!} \prod_{i} \frac{(\rho/2)^{A_{ii}/2} e^{-\rho/2}}{(A_{ii}/2)!} \\ &= \frac{\rho^m e^{-\rho n^2/2}}{\prod_{i<j} A_{ij}! \prod_i A_{ii}!!}, \end{aligned} \tag{1.11}$$

where the double factorial $(2x)!! = 2^x x!$.

In the common sparse setting, where the density $\rho \to 0$, the multigraph model reduces to the simple graph model since it becomes exponentially unlikely to observe more than a single edge between any pair of nodes. Specifically, by setting $\rho = \frac{n}{n-1}p$ (adjusted for self-edges), the multigraph likelihood Eq. (1.11) approximates the simple graph likelihood Eq. (1.2) as

$$P(\boldsymbol{A}|\rho) = P(\boldsymbol{A}|p) + O(p^2) \tag{1.12}$$

to leading order in this limit. This approximation allows us to use the more computationally manageable multigraph model in sparse settings, even for networks known to be simple. Consequently, the Erdős-Rényi multigraph shares the giant component and connectivity properties previously discussed in the sparse, constant degree limit[3].

To complete the description of the multigraph model, we must adopt a prior for the density ρ. Unlike the probability $p \in [0, 1]$, the density ρ can assume any non-negative value. As it is unbounded, a traditional maximum entropy prior is undefined over the infinite range. Instead, we maximize the entropy under the constraint that the expected

[3] Solomonoff and Rapoport [122] initially demonstrated these properties in the multigraph version of the model, while Erdős and Rényi [50] considered the microcanonical $G(n, m)$.

value is 1, resulting in an exponential prior[4]

$$P(\rho) = e^{-\rho}. \tag{1.13}$$

From this choice, the posterior distribution of ρ is

$$\begin{align} P(\rho|\boldsymbol{A}) &= \frac{P(\boldsymbol{A}|\rho)P(\rho)}{P(\boldsymbol{A})} \tag{1.14} \\ &= \frac{\rho^m e^{-\rho(n^2/2+1)}(n^2/2+1)^{m+1}}{m!}. \tag{1.15} \end{align}$$

The peak of this distribution defines the MAP estimate of ρ

$$\hat{\rho}_{\text{MAP}} = \frac{2m}{n^2+2}. \tag{1.16}$$

Note that this value is slightly less than the empirical density Eq. (1.3), reflecting the influence of the prior towards smaller values.

The Bayesian evidence of the model is found by integrating over the density,

$$\begin{align} P(\boldsymbol{A}) &= \int P(\boldsymbol{A}|\rho)P(\rho)d\rho \\ &= \underbrace{\frac{1}{(n^2/2)^m}\frac{m!}{\prod_{i<j} A_{ij}!\prod_i A_i!!}}_{\text{multinomial}}\underbrace{\frac{(n^2/2)^m}{(n^2/2+1)^{m+1}}}_{\text{geometric}} \tag{1.17} \\ &= P(\boldsymbol{A}|m)P(m), \end{align}$$

here organized to illustrate its microcanonical interpretation. The number of edges m is first distributed geometrically, then those edges are multinomially distributed among possible node pairs.

The description length of the Facebook friends network in the multigraph model is then $H(\boldsymbol{A}) = -\log P(\boldsymbol{A}) = 13333.1$, slightly more than the description length $H(\boldsymbol{A}) = 13249.6$ in the simple graph model. Although the simple Erdős-Rényi model compresses the friends network more efficiently due to its restriction to simple networks, we typically prefer the multigraph model in these sparse cases for its ease and flexibility.

We can further use this model to predict which nodes would likely connect to each other in future observations of the network. In the Erdős-Rényi model, only the parameter ρ

[4] This is analogous to how the Boltzmann distribution in Eq. (A.33) maximizes entropy under the average energy constraint.

is used to make predictions. Given an inference of this density from a network, we can predict that a further observation will share the same density. In many contexts, however, we might anticipate that the predicted data will exhibit a different density than our initial observation.

This is especially relevant in a *cross-validation* context, one where the observed data is randomly divided into a training set $\boldsymbol{A}^{\text{train}}$ and testing set $\boldsymbol{A}^{\text{test}}$ such that $\boldsymbol{A}^{\text{train}} + \boldsymbol{A}^{\text{test}} = \boldsymbol{A}$. In these tests, the model is often trained on 80% of the data and tested on the remaining 20%. If we infer that the training data has density ρ^{train}, we can then assume that the testing data only has a quarter the density, setting $\rho^{\text{test}} = f\rho^{\text{train}}$ with $f = 0.25$.

Under this assumption, we can then define a posterior-predictive distribution that accounts for all possible values of the density we could infer, factoring in the ratio f between the training and testing data,

$$\begin{aligned} P(\boldsymbol{A}^{\text{test}}|\boldsymbol{A}^{\text{train}}, f) &= \int P(\boldsymbol{A}^{\text{test}}|\rho^{\text{test}})P(\rho^{\text{train}}|\boldsymbol{A}^{\text{train}})d\rho^{\text{train}} \\ &= \int P(\boldsymbol{A}^{\text{test}}|f\rho)P(\rho|\boldsymbol{A}^{\text{train}})d\rho \\ &= \frac{(m^{\text{train}} + m^{\text{test}})!}{m^{\text{train}}! \prod_{i<j} A_{ij}! \prod_i A_{ii}!!} \frac{f^{m^{\text{test}}}(n^2/2+1)^{m^{\text{train}}+1}}{((1+f)n^2/2+1)^{m^{\text{train}}+m^{\text{test}}+1}}. \end{aligned} \tag{1.18}$$

As in this example, the posterior-predictive can be quite complicated. For the other models we discuss, the predictive is detailed in Appendix B.8. By averaging results over many cross-validation splits, we then obtain an alternative perspective on the model's performance, specifically how well it can reconstruct the full data set from partial observations. In our network of friendships, we can compute that on average the *minus log posterior-predictive* is

$$H(\boldsymbol{A}^{\text{test}}|\boldsymbol{A}^{\text{train}}, f) = -\log P(\boldsymbol{A}^{\text{test}}|\boldsymbol{A}^{\text{train}}, f) \approx 3670.2, \tag{1.19}$$

indicating how effectively the model can extrapolate the network.

As shown in Appendix A.4, suitably interpreted this measure of predictive power is very similar to the model evidence criterion, although subtle differences exist between the two. When assessing models on real network data we will report both these statistics in this thesis.

1.2.2 Configuration model

While the Erdős-Rényi random graph provides key insights into network connectivity, its structureless nature limits its ability to express further features of real networks. To address this, we employ models that reflect commonly observed network structures. In this section we describe the *configuration model*, which accounts for the wide variation in node degrees often seen in real examples.

In many networks the distribution of connections is uneven; some nodes have many interactions, while most only have a few. Figure 1.7a revisits the social network from Figure 1.5, now with node sizes representing node degree. These degrees illustrate that the network features a central core of highly connected nodes alongside peripheral nodes with fewer connections, resulting in the broad degree distribution plotted in Figure 1.7c. In this section, we describe the usual configuration model of this variation and introduce a more general configuration model that allows us to quantify this inequality.

To demonstrate how significant this observed variation is, we can compare the social network to random graphs with the same number of nodes and edges. Figure 1.7b contains such a random network, which apparently has a more homogeneous degree distribution than the observations. As depicted in Figure 1.7c, no node in the random graph has more than 24 connections, whereas a node in the real network reaches a maximum degree $k_{\max} = 58$. This is not an isolated example. In Figure 1.7d we plot the maximum degrees $k_{\max}$ observed across 1,000 random graphs sampled from the microcanonical $G(m, n)$. In no case does a node have degree over 32. This discrepancy indicates that while it is technically possible for the flat random graph to generate a network as skewed as our observation, the p-value is exceedingly low. We therefore reject the Erdős-Rényi model for this data.

To bridge this gap we must add something to the Erdős-Rényi model. If everyone has the same opportunity ρ to connect with everyone else, it is unlikely that the variation in degrees we observe could appear. To create a model which can generate such an imbalance, we instead assume that the nodes are not equivalent. We assign to each node i a propensity (or "node weight") $\theta_i \geq 0$ to connect to other nodes, organized as entries of an n-vector $\boldsymbol{\theta}$. Nodes i and j are then connected by $\theta_i \theta_j \rho$ edges on average, the product of their individual weights and the overall density ρ. Figure 1.8a shows a schematic of this canonical generative process in action.

Since the node weights always appear multiplied by the global density, their overall normalization is arbitrary since it can be compensated by a fluctuation of the density parameter ρ. We therefore choose to normalize the node weights to have average value 1

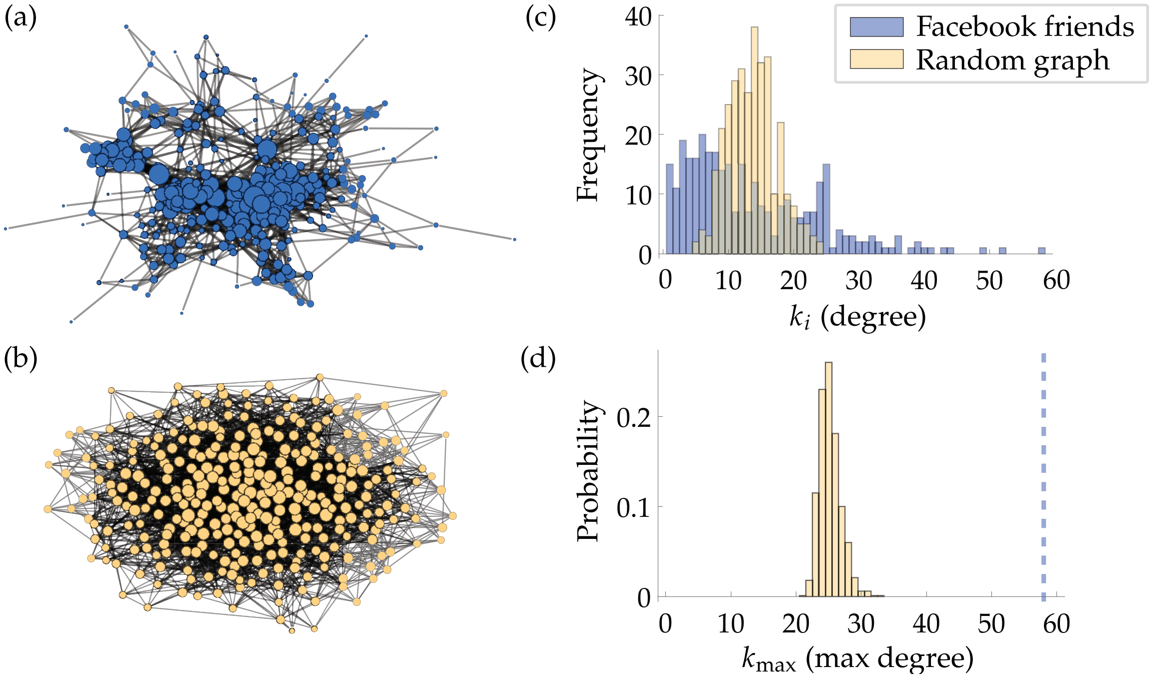


Figure 1.7: (a) Social network of Facebook profile connections, nodes sized according to their degree [17]. (b) A sampled Erdős-Rényi random graph with the same number of nodes and edges as the social network. (c) Histogram of degree distributions of the social network (blue) and random graph (orange). (d) Distribution of the maximum degree k_{max} in the random graph null model, as observed over 1,000 samples. The maximum degree of 58 achieved by the social network is highlighted in blue and is well-separated from that obtained by a typical random graph.

as

$$\frac{1}{n}\sum_{i=1}^{n}\theta_i = 1. \tag{1.20}$$

This choice constrains the parameters to the scaled simplex $n\Delta_{n-1} \subset \mathbb{R}^n$.

Extending the Erdős-Rényi random multigraph Eq. (1.11), we can then write the overall likelihood as a product of Poisson distributions

$$\begin{aligned} P(\boldsymbol{A}|\boldsymbol{\theta},\rho) &= \prod_{i<j}\frac{(\theta_i\theta_j\rho)^{A_{ij}}e^{-\theta_i\theta_j\rho}}{A_{ij}!}\prod_{i=1}^{n}\frac{(\theta_i^2\rho/2)^{A_{ii}/2}e^{-\theta_i^2\rho/2}}{(A_{ii}/2)!} && (1.21)\\ &= \frac{\rho^m e^{-n^2\rho/2}}{\prod_{i<j}A_{ij}!\prod_i A_{ii}!!}\prod_{i=1}^{n}\theta_i^{k_i}. && (1.22)\end{aligned}$$

This likelihood defines the canonical configuration model, also known as the Chung-Lu model[5] [30]. In this model the node weight θ_i controls the expected degree of node i as

$$\mathbf{E}k_i = \sum_{j=1}^{n}\mathbf{E}A_{ij} = \sum_{j=1}^{n}\theta_i\theta_j\rho = \theta_i n\rho. \tag{1.23}$$

As we vary these node weights we thus expect to generate a network with a realistic range of degrees.

To perform Bayesian inference with this model we must specify priors over the parameters. For the global weight ρ we use the same exponential prior Eq. (1.13) as the Erdős-Rényi multigraph. For the node weights $\boldsymbol{\theta}$ we adopt a uniform, maximum entropy prior over the simplex of values satisfying the normalization Eq. (1.20),

$$P(\boldsymbol{\theta}) = \frac{(n-1)!}{n^{n-1}}. \tag{1.24}$$

Integrating over this simplex yields an *integrated likelihood*

$$\begin{aligned} P(\boldsymbol{A}|\rho) &= \int P(\boldsymbol{A}|\boldsymbol{\theta},\rho)P(\boldsymbol{\theta})d\boldsymbol{\theta}\\ &= \frac{\prod_i k_i!}{(2m)!\prod_{i<j}A_{ij}!\prod_i A_{ii}!!}\binom{2m+n-1}{n-1}^{-1}\rho^m e^{-n^2\rho/2}\end{aligned}$$

[5] Some presentations of the configuration model use probabilities proportional to the observed degrees rather than the latent node weights. This substitution can be understood as an "empirical Bayes" treatment of this Chung-Lu model.

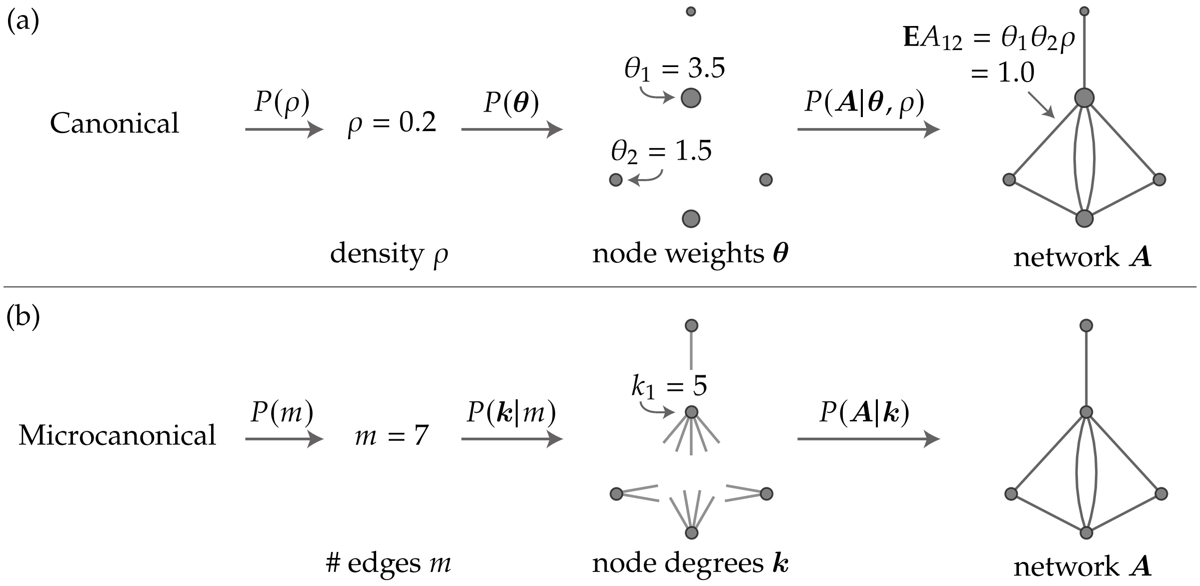


Figure 1.8: Schematic of perspectives on the generative process in the configuration model. (a) Canonically, the overall density ρ is first sampled from the prior $P(\rho)$, followed by the node weights $\boldsymbol{\theta}$ indicated by the node sizes, and then the network $\boldsymbol{A}$ itself is generated with edges appearing proportionally to the weights they connect. Example parameters ρ, θ_1, θ_2 relevant to generating A_{12} are given. (b) Microcanonically, the number of edges m is first set, followed by a degree sequence $\boldsymbol{k}$ that sums to $2m$. Finally the network is generated by a stub matching process that wires together the edges.

which we can then further integrate over ρ to obtain the evidence

$$\begin{aligned} P(\boldsymbol{A}) &= \int P(\boldsymbol{A}|\rho)P(\rho)d\rho \\ &= \underbrace{\frac{(2m)!!\prod_i k_i!}{(2m)!\prod_{i<j}A_{ij}!\prod_i A_{ii}!!}}_{\text{stub-matching}} \underbrace{\binom{2m+n-1}{n-1}^{-1}}_{\text{uniform}} \underbrace{\frac{(n^2/2)^m}{(n^2/2+1)^{m+1}}}_{\text{geometric}} \\ &= P(\boldsymbol{A}|\boldsymbol{k})P(\boldsymbol{k}|m)P(m) \end{aligned} \tag{1.25}$$

where we have highlighted the microcanonical structure of the model. In this decomposition the number of edges m is distributed geometrically, as in the Erdős-Rényi multigraph Eq. (1.17). The degree sequence $\boldsymbol{k}$ is then uniformly distributed among all possible vectors of n non-negative integers that sum to $2m$. Finally given these degrees, the network itself is distributed according to a so-called *stub-matching* likelihood $P(\boldsymbol{A}|\boldsymbol{k})$.

This network likelihood corresponds to an underlying stub-matching process. In it, we first assign k_i "stubs" (or "half-edges") to each node i. To form each edge of the network we randomly select two stubs which we "wire together" by replacing them with an edge that connects the two nodes. By performing this matching m times we form all edges of the network $\boldsymbol{A}$. This stub matching generically generates multigraphs as self edges can appear and two nodes can be connected multiple times.

To recover the likelihood of generating a particular multigraph $\boldsymbol{A}$ in this manner, we first note that if all stubs are distinguishable there are

$$\Omega(\boldsymbol{M}) = \frac{(2m)!}{(2m)!!} \tag{1.26}$$

possible matchings (or "perfect matchings") of the $2m$ stubs. As the stubs are ultimately indistinguishable, a multiplicity

$$\Xi(\boldsymbol{A}) = \frac{\prod_i k_i!}{\prod_{i<j}A_{ij}!\prod_i A_{ii}!!} \tag{1.27}$$

of these matchings correspond to the same graph. As we uniformly select one of the $\Omega(\boldsymbol{M})$ possible stub matchings, the probability of generating a particular network $\boldsymbol{A}$ with degrees $\boldsymbol{k}$ is thus

$$\begin{aligned} P(\boldsymbol{A}|\boldsymbol{k}) &= \frac{\Xi(\boldsymbol{A})}{\Omega(\boldsymbol{M})} \\ &= \frac{(2m)!!\prod_i k_i!}{(2m)!\prod_{i<j} A_{ij}!\prod_i A_{ii}!!}. \end{aligned} \tag{1.28}$$

This stub-matching likelihood alone is often also referred to as the "configuration model" since it is a model over possible networks that share exactly the same degree sequence $\boldsymbol{k}$. In Section 1.2.3 this microcanonical model will prove useful as a null model for demonstrating group structure.

Concluding with this stub-matching, the full microcanonical perspective on network formation in the configuration model is given in Figure 1.8b alongside the canonical picture. Both formulations lead to the same overall likelihood of generating a given network and therefore may be adopted interchangeably. An advantage of the microcanonical form in this case is an explicit expression $P(\boldsymbol{k}|m)$ for the broad distribution of degree sequences realized by the configuration model.

Although it is a common feature, not all networks have such variation in their degree distribution. In Figure 1.9a we plot the degree distributions of both the earlier Facebook friends network and the college football matches described in Section 1.1.2. Although there is some variation in the degrees of the football network, they are much more consistent than in the Facebook network, as each team plays roughly the same number of matches over the course of the season. This raises a natural model selection problem; is the level of variation of degrees we observe in the football network sufficient to justify a generalization like the configuration model? Or should we prefer the simpler Erdős-Rényi model?

To help answer this question we could repeat the null model significance tests for the football network, and check whether the Erdős-Rényi model can be rejected. Alternatively, we can compare the Bayesian evidence of the two models, Eq. (1.17) for the Erdős-Rényi model and Eq. (1.25) for the configuration model. Table 1.1 contains these results, expressed as description lengths $H(\boldsymbol{A})$ for the two models on the two data sets. From these values we can observe that the configuration model more efficiently encodes the social network while the Erdős-Rényi model better encodes the football network. By using these description lengths we thus establish affirmative evidence of one model over another, not just reject or fail to reject the null Erdős-Rényi model.

An alternative approach to comparing these evidences is to instead define a *nested model* that includes both the Erdős-Rényi and configuration models as special cases. To accomplish this we note that if all node weights are the same, $\theta_i = 1$, the likelihood

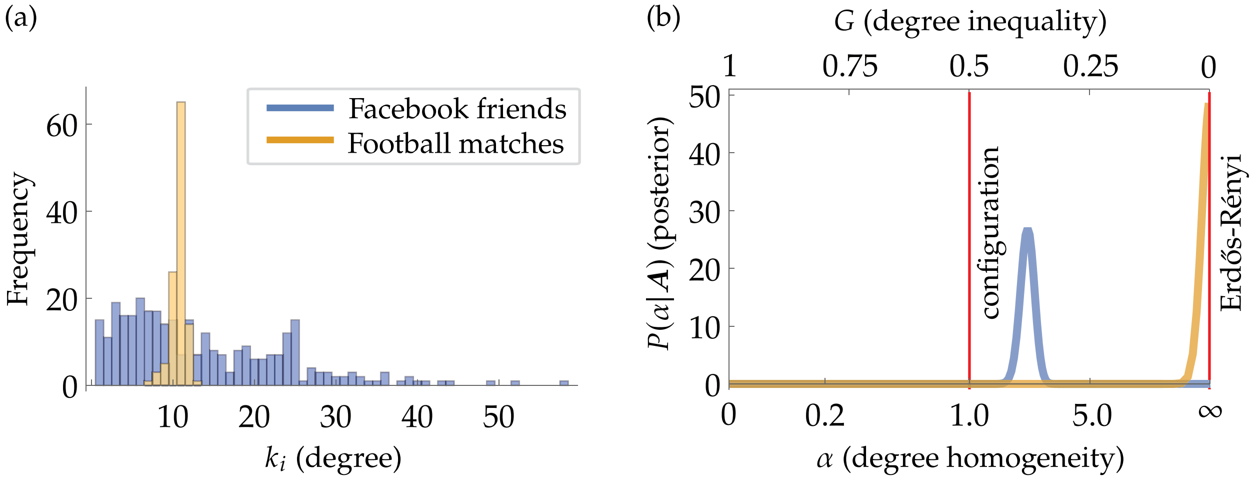


Figure 1.9: (a) Degree distributions of the Facebook friend and football match networks. (b) Posterior distributions of the degree homogeneity parameter α in the general configuration model on these networks. Values of α are plotted along the bottom x-axis and corresponding values of the Gini coefficient of degree inequality G are given on the top x-axis. The overall scale is linear in the Gini coefficient, leading to a Jacobian factor in the posterior such that the plot reflects the appropriate density. The configuration model is highlighted by the vertical line at $\alpha = 1, G = 0.5$, along with the Erdős-Rényi model at $\alpha \to \infty, G = 0$.

Data set	Facebook friends			College football		
Model	E-R	config.	general	E-R	config.	general
$\hat{\alpha}_{\text{MAP}}$	N/A	N/A	1.926	N/A	N/A	271.3
$\hat{G}_{\text{MAP}}$	N/A	N/A	0.381	N/A	N/A	0.034
$H(\boldsymbol{A})$	13333.1	12334.8	**12299.7**	**3004.8**	3212.88	3014.6
$H(\boldsymbol{A}^{\text{test}}\vert\boldsymbol{A}^{\text{train}})$	3670.2	3420.1	**3415.6**	**859.0**	890.1	859.6

Table 1.1: Table of performance metrics of the Erdős-Rényi, configuration, and general configuration models on the Facebook friend and College football network examples. The Bayesian evidence $H(\boldsymbol{A})$, minus log posterior-predictive $H(\boldsymbol{A}^{\text{test}}|\boldsymbol{A}^{\text{train}})$, and best-fit parameter $\hat{\alpha}_{\text{MAP}}$ in the general model and corresponding Gini coefficient $\hat{G}_{\text{MAP}}$ are given. The log posterior-predictive represents the median result out of 50 cross-validation splits, as described in Appendix B.8.

Eq. (1.21) is equal to the Erdős-Rényi model likelihood Eq. (1.11). This case can be represented by a Dirac-delta prior on the weights

$$P(\boldsymbol{\theta}) = \prod_{i=1}^{n} \delta(\theta_i - 1). \tag{1.29}$$

To interpolate between this and the uniform prior Eq. (1.24) that defines the configuration model, we use a *Dirichlet* prior[6] with concentration parameter $\alpha > 0$ over the simplex

$$P(\boldsymbol{\theta}|\alpha) = \frac{\Gamma(n\alpha)}{\Gamma(\alpha)^n n^{n-1}} \prod_{i=1}^{n} \theta_i^{\alpha-1}. \tag{1.30}$$

When $\alpha = 1$, the Dirichlet distribution is the uniform Eq. (1.24), while in the limit $\alpha \longrightarrow \infty$ the distribution converges to the Dirac-delta Eq. (1.29). Therefore the same model likelihood with this now variable prior includes both the usual Erdős-Rényi and configuration models as special cases. We will call this broader model with the extra parameter α the *general configuration model.*

Integrating over both the node weights $\boldsymbol{\theta}$ and density ρ in the general model, we obtain its microcanonical form

$$\begin{aligned} P(\boldsymbol{A}|\alpha) &= \underbrace{\frac{(2m)!! \prod_i k_i!}{(2m)! \prod_{i<j} A_{ij}! \prod_i A_{ii}!!}}_{\text{stub-matching}} \underbrace{\binom{2m + n\alpha - 1}{n\alpha - 1}^{-1} \prod_{i=1}^{n} \binom{k_i + \alpha - 1}{\alpha - 1}}_{\text{Dirichlet-multinomial}} \underbrace{\frac{(n^2/2)^m}{(n^2/2 + 1)^{m+1}}}_{\text{geometric}} \\ &= P(\boldsymbol{A}|\boldsymbol{k})P(\boldsymbol{k}|m, \alpha)P(m). \end{aligned} \tag{1.31}$$

Here the degree sequence follows a *Dirichlet-multinomial* distribution $P(\boldsymbol{k}|m, \alpha)$, and the binomial coefficient is generalized to possibly non-integer arguments as

$$\binom{a}{b} = \frac{\Gamma(a+1)}{\Gamma(b+1)\Gamma(a-b+1)}. \tag{1.32}$$

This is the discrete analog of the Dirichlet distribution, defined over possible non-negative integer sequences $\boldsymbol{k}$ that sum to $2m$. Like the Dirichlet distribution, when $\alpha = 1$ the Dirichlet-multinomial distribution is uniform over all possible degree sequences. As α increases, the degree distribution $\boldsymbol{k}$ is increasingly concentrated. In the limit $\alpha \longrightarrow \infty$ the distribution of the degrees is multinomial, recovering the Erdős-Rényi model. Between

[6] Dirichlet distributions are typically defined over the unit simplex. We include the scale factor n^{n-1} in this presentation to account for the normalization Eq. (1.20).

these values, $1 < \alpha < \infty$, the degree distribution interpolates between the homogeneous Erdős-Rényi distribution to the dispersed configuration model. And for $0 < \alpha < 1$, the degrees vary even more than in the configuration model.

As larger values of α produce more uniform degree distributions, we refer to this α as the *degree homogeneity* parameter. By fitting this parameter to a network we can thus infer its degree homogeneity. To do this in the Bayesian framework, we introduce a half-Cauchy prior

$$P(\alpha) = \frac{2}{\pi(\alpha^2 + 1)} \tag{1.33}$$

which has the special case $\alpha = 1$ as its median value. Including this prior, our model has become too complicated to analytically perform the integral over the parameter $\alpha > 0$, although we can numerically integrate to obtain the evidence of the general model as

$$P(\boldsymbol{A}) = \int P(\boldsymbol{A}|\alpha)P(\alpha)d\alpha. \tag{1.34}$$

As the models we consider become increasingly complex, we will need to resort to such tricks to compute the model evidence.

To help interpret the degree homogeneity α, we can instead re-frame it as a measure of *degree inequality*. Inequality of a resource distribution, for example of an income distribution, is often quantified using the *Gini coefficient* [82, 56]. This measure ranges from 0 when all values are same, to 1 when only one value is non-zero. In our case, we can consider the node weights $\boldsymbol{\theta}$ to be a resource and so quantify the inequality of their distribution amongst the nodes[7]. The Erdős-Rényi case where all weights are the same, $\theta_i = 1$, corresponds to a Gini of 0. For the more general Dirichlet distribution Eq. (1.30), we can compute that the Gini index of the θ_i is

$$G = \frac{\Gamma(\alpha + 1/2)}{\Gamma(\alpha + 1)\Gamma(1/2)} = \binom{\alpha - 1/2}{\alpha}. \tag{1.35}$$

For the usual configuration model, $\alpha = 1$, this Gini is $G = 0.5$. In the extreme limit $\alpha \to 0$, the Gini is maximized at $G = 1$ as only one node has non-zero degree. Table 1.2 summarizes these special cases of the degree homogeneity and inequality parameters. In Figure 1.9b we also plot the homogeneity parameter α along the bottom x-axis and the Gini G along the top x-axis to highlight the inverse relation between the two.

[7] Here we define the Gini coefficient in terms of the distribution of the latent node weights $\boldsymbol{\theta}$. Most uses of the Gini instead consider the distribution of an observed resource like the actual realized degrees $\boldsymbol{k}$.

Model	α	G
Erdős-Rényi random graph	∞	0
Configuration model	1	1/2
General configuration model	inferred	inferred

Table 1.2: Summary of the parameter values of the general configuration model that correspond to the other models it generalizes. The degree homogeneity α and corresponding degree inequality G are inferred in the general configuration model.

Since the nested model includes these special cases of interest, the posterior distribution can be used to directly compare their model evidence as

$$\frac{P_{\text{ER}}(\boldsymbol{A})}{P_{\text{config}}(\boldsymbol{A})} = \frac{P(\boldsymbol{A}|\alpha \to \infty)}{P(\boldsymbol{A}|\alpha = 1)} = \frac{P(\alpha \to \infty|\boldsymbol{A})}{P(\alpha = 1|\boldsymbol{A})} \frac{P(\alpha = 1)}{P(\alpha \to \infty)}. \tag{1.36}$$

In Figure 1.9, the posterior distributions are plotted including the prior factor $\frac{P(\alpha=1)}{P(\alpha\to\infty)}$, so that we can visually inspect whether one model is preferred over another in this manner[8]. We can observe that the posterior for the football match network is peaked as $\alpha \to \infty$, indicating that the Erdős-Rényi random model is not excluded. For the social network, the posterior is peaked at $\alpha \approx 1.93$, between the configuration and Erdős-Rényi models. Although the relative proximity to the configuration model indicates that it is preferred over the Erdős-Rényi model, both special cases are excluded in favor of the more general model.

In Table 1.1, the full model evidence and predictive power of the general model confirm these findings. The Facebook network is best described by the general configuration model while the football network is best captured by the Erdős-Rényi model. The general configuration model over-fits the football data and the extra flexibility only diminishes the predictive power of the Erdős-Rényi model. For many of the models that we consider later in this thesis, the absolute evidence is difficult to compute. By constructing such nested models, we can straightforwardly observe relative evidence of the models it generalizes by whether the posterior distribution of interpolating parameters excludes those special cases.

At this stage it is worth reflecting that the three models we have considered in this section – the Erdős-Rényi, configuration, and general configuration models – all share the likelihood Eq. (1.21) and are only differentiated by the form of the prior on the node weights. Despite the commonality, the models exhibit drastically different results. This

[8] In this thesis when using a non-uniform prior like $P(\alpha)$, we will rescale the axis such that the appropriate density reflects the prior factor, allowing us to interpret the results in this manner.

behavior is unlike many statistical settings where as we collect more data the influence of the prior wanes, such as in Appendix A.1. This intuition, however, breaks down for network models where we associate to each node i a new parameter like θ_i. In these situations, as the size of the network increases, $n \to \infty$, the number of parameters $\boldsymbol{\theta}$ grows as well. The ratio between the overall quantity of data, roughly the number of edges m, and the number of parameters n stays fixed in the usual constant degree limit. The prior on these node-level parameters therefore significantly influences model outcomes. In contrast, priors like $P(\rho)$ on single global parameters still have little bearing in large networks, since the influence of the single prior does not scale with the network size.

Much of this thesis follows a similar theme to this observation. In realistic settings, terms which may appear to be "subleading" in a traditional sense, like choice of prior, can in fact matter considerably for real applications. By carefully including these details in our analysis we can obtain not only more faithful results but often also learn about the underlying nature of the data sets.

1.2.3 Stochastic block model

In the previous section, we extended the Erdős-Rényi random graph to the general configuration model to detect and quantify statistically significant variations in node degrees. In this section, we introduce the stochastic block model (SBM), which similarly models group structures such as those discussed in Section 1.1.2.

Groups in networks are often characterized as communities of tightly connected nodes. In the examples shown in Figure 1.3a-c, the groups are assortative as indicated by the high number of edges m_{in} between nodes of the same group. To count these intra-group edges, we index the groups by $r = 1, \ldots, q$ and represent group assignments as an n-vector of integers $\boldsymbol{b}$, where each node i belongs to group $b_i \in \{1, \ldots, q\}$. In this notation, the number of edges inside groups is

$$m_{\text{in}} = \frac{1}{2} \sum_{i,j=1}^{n} A_{ij} \delta_{b_i b_j}, \tag{1.37}$$

where the Kronecker delta $\delta_{b_i b_j}$ restricts the sum to nodes i and j in the same group. While a large number of edges m_{in} within the groups suggests a strong group structure, it is important to contextualize this count. Even if there is no assortative preference, randomly placed edges can fall within groups and contribute to m_{in}.

To establish a baseline for m_{in}, we use the microcanonical configuration model Eq. (1.28) as a null hypothesis. This generates alternative networks that match the observed node degrees but lack inherent group structure. Under this randomization, the expected number of edges between any two nodes i and j is

$$\mathbf{E}A_{ij} = \frac{k_i k_j}{2m}. \tag{1.38}$$

We thus expect a total number of intra-group edges

$$\langle m_{\text{in}} \rangle_{\text{config}} = \frac{1}{2} \sum_{ij} \frac{k_i k_j}{2m} \delta_{b_i b_j}. \tag{1.39}$$

To demonstrate that the groups are meaningfully assortative, we then check if the count m_{in} observed in the network is surprising relative to this expectation.

A measure known as the *modularity* [98] quantifies and normalizes this difference as

$$\begin{aligned} Q(A, \boldsymbol{b}) &= \frac{1}{m} \left(m_{\text{in}} - \langle m_{\text{in}} \rangle_{\text{config}} \right) \\ &= \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta_{b_i b_j}. \end{aligned} \tag{1.40}$$

This definition ensures that the expected modularity of a partition is 0, and a positive value indicates that the network is more assortative, or *modular*, than we would expect.

Within the significance testing framework, we can also compute the p-value that the configuration model could generate a network as assortative as the observed case. Figure 1.10 shows an example of this test on the network of Division IA college football matches considered in the previous section. Here we can calculate the modularity of the network across the $q = 12$ conferences, represented by node colors in Figure 1.10a. Across the season, $m_{\text{in}} = 397$ of the $m = 616$ matches are played between teams in the same conference (highlighted in green), leading to a network modularity of $Q = 0.555$. Figure 1.10b compares the modularity along this partition in the real network to that in 10,000 alternative networks drawn from the configuration model. Across these cases the largest modularity obtained is $Q = 0.041$, indicating that there is a very statistically significant preference for teams to play within their own conference.

In an unsupervised setting, where the group structure of the network is unknown, the modularity is often used not only to measure behavior but also to identify groups. In this approach, a "good" group structure is defined as one with high modularity, where groups contain significantly more internal edges than expected by chance. Thus, for a

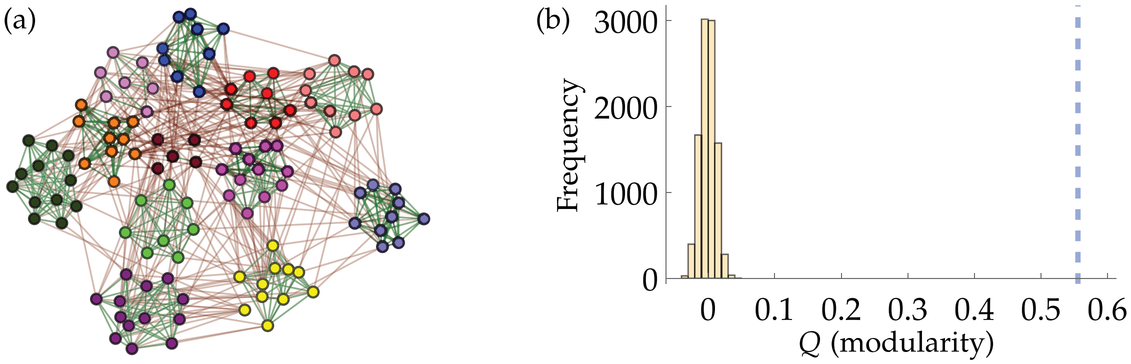


Figure 1.10: (a) Football match network with edges within groups highlighted in green and edges between groups in red. (b) Modularity of the conference partition of the football match network (vertical dashed line) compared to 10,000 alternative networks with the same degree sequence sampled from the configuration model.

given network $\boldsymbol{A}$, the best-fit group structure is the optimum

$$\boldsymbol{b}^* = \underset{\boldsymbol{b}}{\operatorname{argmax}}\, Q(\boldsymbol{A}, \boldsymbol{b}), \tag{1.41}$$

where $\boldsymbol{b}^*$ is found using one of several *modularity maximization* methods. Some commonly used algorithms are discussed in Appendix C.16.2.

Applying this strategy to the football match network, we can find a partition of the teams into 11 groups with modularity $Q = 0.601$. Remarkably, this partition closely resembles the "true" conference group structure. Using the information-theoretic similarity measure defined in Chapter 3, the found partition scores a 0.865 out of 1. From the network alone, we can thus nearly recover the original conferences. Encouraged by this result, we can apply this method to help uncover groups of nodes that meaningfully influence the structure of the network, even when such groups are initially unknown.

While this modularity maximization approach has been widely used to great effect in network science [13, 19, 58], it has certain limitations. One major issue is overfitting. Consider a graph that inherently lacks a group structure, such as those generated by Erdős-Rényi or configuration models. In such cases the appropriate "partition" of the nodes places them into one large group, $\boldsymbol{b} = (1, \ldots, 1)$. From the definition of the modularity, this all-in-one grouping has $Q = 0$ for any network $\boldsymbol{A}$. However, when optimizing the modularity, it is typically possible to find some other partition of the nodes into more than one group that is at least slightly assortative, $Q > 0$, merely due to random graph fluctuations. Modularity maximization will therefore prefer this over-fitted partition over the true single group.

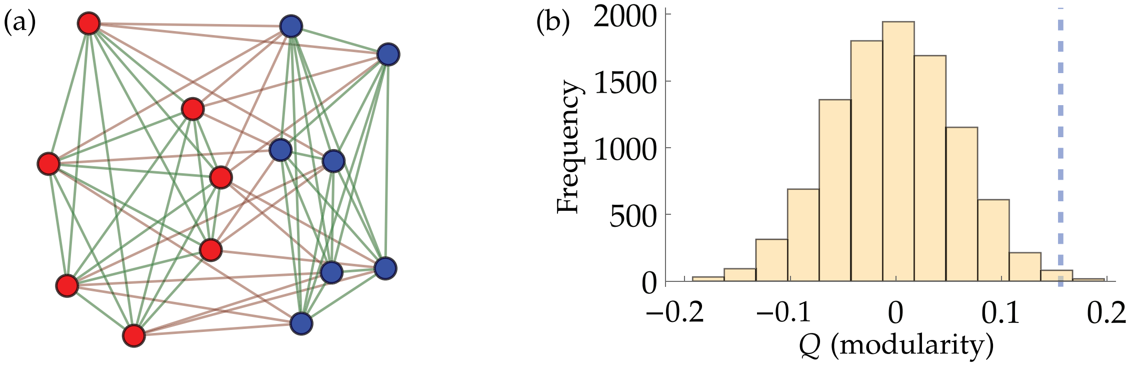


Figure 1.11: (a) Network of football matches within only the Big Ten conference. The teams are partitioned into two groups to maximize modularity. (b) The modularity along the optimized partition compared that in 10,000 networks with the same degree sequence sampled from the configuration model.

Figure 1.11 exemplifies this issue by illustrating the network of matches played within only the Big Ten college football conference in 2022. This network presumably lacks group structure, given that it represents a single conference. Yet, the modularity is maximized at positive $Q = 0.156$ by dividing the teams into two groups, as shown in Figure 1.11a. In Figure 1.11b we again consider this modularity in the context of 10,000 alternative networks that share the same degree sequence. In this case, the observed modularity is not clearly separated from what the configuration model predicts, but still yields a p-value $P = 0.01$. In isolation, this might imply a statistically significant assortative pattern. However, given that this grouping was selected among $2^{n-1} - 1 = 8191$ possible two-group partitions, is it not unexpected that one might exhibit such a low p-value. Since the modularity alone can never return a null result, modularity maximization must be combined with such significance testing that is often challenging to interpret.

Even when no true assortative pattern is present, modularity maximization will overfit and report some slightly assortative partition. In cases where the actual group structure is *disassortative*, such as the bipartite network of Figure 1.3d, modularity maximization will also fail to identify the true groups. This is because disassortative groups do not align with modularity's fundamentally assortative definition of a community. To successfully identify such groups, we must broaden our definition to include these cases, or even more complex structural patterns like mixtures of assortative and disassortative groups.

The root of these issues lies in the fact that modularity maximization is not a generative model, unlike the Erdős-Rényi or configuration models described earlier. While the partition that maximizes the modularity is useful for describing and summarizing assortative network structures, it does not provide a mechanism for their formation. This precludes us from employing our usual Bayesian inference tools to prevent overfitting or from using the modularity to make predictions.

While some efforts have been made to directly convert the modularity objective function into a generative model [110], in this thesis we instead consider *stochastic block models* (SBMs). These models come in a variety of different flavors [69, 106, 141], some of which we introduce in Chapter 2. They provide the flexibility to model a wide range of possible group structures and can be directly compared against the other generative models we have considered.

The traditional stochastic block model is defined by the assumption that the probability two nodes i and j are connected depends only on their group identities b_i and b_j. Some pairs of groups may be more likely to be connected than others, but all nodes within the same group share the same structural role: they are identically and independently likely to be connected to nodes in other groups. These probabilities across groups form a symmetric $q \times q$ *weight matrix* $\boldsymbol{\omega}$, where ω_{rs} is the expected number of edges between each node in group r and each node in group s. This framework effectively defines what we mean by groups: node labels that influence the pattern of connections. This pattern can be quite generic as we do not assume the groups are defined by a globally assortative or disassortative preference.

For each pair of groups r and s, the weight matrix entry ω_{rs} plays the same role as the overall density ρ in the Erdős-Rényi multigraph model Eq. (1.11). Each edge count A_{ij} is modeled as a Poisson distribution with expectation $\omega_{b_i b_j}$. Consequently, the usual SBM models the interior of each group r as a random multigraph with density ω_{rr} and assumes connections between groups occur independently and identically.

By building the model in this way, the SBM becomes a nested model that includes an Erdős-Rényi random graph as the special case where all nodes are assigned to a single group, $\boldsymbol{b} = (1, ..., 1)$. Just as the general configuration model, this nested structure allows us to directly compare the SBM against the Erdős-Rényi model it extends. Collecting these assumptions, the likelihood that a network is generated by a group structure $\boldsymbol{b}$ and weights $\boldsymbol{\omega}$ in the SBM is then

$$P(\boldsymbol{A}|\boldsymbol{\omega}, \boldsymbol{b}) = \prod_{i<j} \frac{\omega_{b_i b_j}^{A_{ij}} e^{-\omega_{b_i b_j}}}{A_{ij}!} \prod_{i=1}^{n} \frac{(\omega_{b_i b_i}/2)^{A_{ii}/2} e^{-\omega_{b_i b_i}/2}}{(A_{ii}/2)!}. \tag{1.42}$$

We can condense this expression by introducing some useful notation. We denote the number of nodes in group r as

$$n_r = \sum_{i=1}^{n} \delta_{b_i r}, \tag{1.43}$$

forming the q-vector of integers $\boldsymbol{n}$, and count the number of edges that run between groups $r, s = 1, ..., q$ as

$$M_{rs} = \sum_{i,j=1}^{n} A_{ij} \delta_{b_i r} \delta_{b_j s}, \tag{1.44}$$

entries of the symmetric $q \times q$ *edge count matrix* $\boldsymbol{M}$. Analogous to the adjacency matrix, the diagonal elements of this edge count matrix M_{rr} are twice the number of edges that run internally between nodes in group r. This convention ensures that the row sum of the edge count matrix $\boldsymbol{m}$ has entries

$$m_r = \sum_{s=1}^{q} M_{rs} = \sum_{i=1}^{n} k_i \delta_{b_i r} \tag{1.45}$$

equal to the total degree of the nodes in each group. With this notation we can then collect the likelihood terms as

$$P(\boldsymbol{A}|\boldsymbol{\omega}, \boldsymbol{b}) = \frac{1}{\prod_{i<j} A_{ij}! \prod_i A_{ii}!!} \prod_{r<s} \omega_{rs}^{M_{rs}} e^{-n_r n_s \omega_{rs}} \prod_{r=1}^{q} \omega_{rr}^{M_{rr}/2} e^{-n_r^2 \omega_{rr}/2}. \tag{1.46}$$

At this stage, we may be tempted to perform a maximum-likelihood estimate and find the choice of weights $\boldsymbol{\omega}$ and partition $\boldsymbol{b}$ most likely to produce the observed network. Unfortunately this approach has serious issues. Consider the group partition $\boldsymbol{b} = (1, ..., n)$ that places each node in its own group and the weight matrix $\boldsymbol{\omega} = \boldsymbol{A}$ between the now n groups. Although this choice of parameters then has a very high model likelihood Eq. (1.42), as each edge A_{ij} is drawn from a Poisson distribution of the same mean A_{ij}, the overall model is woefully over-parametrized. The weight matrix alone contains as many parameters as the data $\boldsymbol{A}$ itself and overfits the network.

Our familiar solution to this problem is to carefully introduce priors over the parameters $\boldsymbol{\omega}$ and $\boldsymbol{b}$ that reflect what we expect "typical" weight and group structures to look like. As in the general configuration model, the choices of these priors matter considerably in realistic network applications. For example in Section 2.2 we discuss how the choice of weight matrix prior $P(\boldsymbol{\omega})$ drastically influences model behavior. In this introduction, however, we will review the choices most often made for the traditional SBM.

Over possible group structures $\boldsymbol{b}$ we use a prior

$$\begin{aligned} P(\boldsymbol{b}) &= P(\boldsymbol{b}|\boldsymbol{n})P(\boldsymbol{n}|q)P(q) \\ &= \frac{\prod_r n_r!}{n!}\binom{n-1}{q-1}^{-1}\frac{1}{n} \end{aligned} \tag{1.47}$$

which is uniform over the number of groups q ranging from 1 to n, over the possible node count vectors $\boldsymbol{n}$ as positive integer vectors of length q that sum to n, and over the possible partitions $\boldsymbol{b}$ that satisfy the node counts $\boldsymbol{n}$. This structure of the prior ensures that *a priori* we have no preference for any particular number of communities q. For instance there is prior probability $P(q = 1) = 1/n$ that all nodes belong in one group. This case recovers the Erdős-Rényi random graph, the possibility of no community structure. If we had instead used a prior that is simply uniform over all possible n-vectors of integers from 1 to n, the prior would heavily weigh a large number of communities, as most such labelings have a number of distinct groups $q \sim n$, and $q = 1$ would be effectively excluded from the prior distribution.

For the weight matrix $\boldsymbol{\omega}$, we traditionally use i.i.d. exponential priors of mean $\rho > 0$ on the upper triangular entries ω_{rs} where $r \leq s$. To generate a symmetric matrix, we then set the entries below the diagonal to those above it as $\omega_{sr} = \omega_{rs}$. This gives the prior over symmetric weight matrices

$$P(\boldsymbol{\omega}|\rho) = \prod_{r\leq s} \frac{1}{\rho} e^{-\omega_{rs}/\rho}. \tag{1.48}$$

We call ρ the *density* parameter here since it is equal to the expected network density averaged over both the likelihood and the weight matrix prior as

$$\mathbf{E}\frac{2m}{n^2} = \frac{1}{n^2}\mathbf{E}\sum_{ij} A_{ij} = \frac{1}{n^2}\sum_{ij}\mathbf{E}\omega_{b_i b_j} = \rho. \tag{1.49}$$

If the number of edges m is known, this density parameter is often set to its empirical point estimate $\hat{\rho} = 2m/n^2$. However, in keeping with our fully Bayesian presentation we instead allow the parameter to run free with the same exponential prior used for the Erdős-Rényi graph, $P(\rho) = e^{-\rho}$.

An example of this generative process is shown in Figure 1.12. Although we start as in the Erdős-Rényi model by setting the overall density ρ, the entries of the weight matrix $\boldsymbol{\omega}$ can then fluctuate from this expectation, producing a network structure differentiated

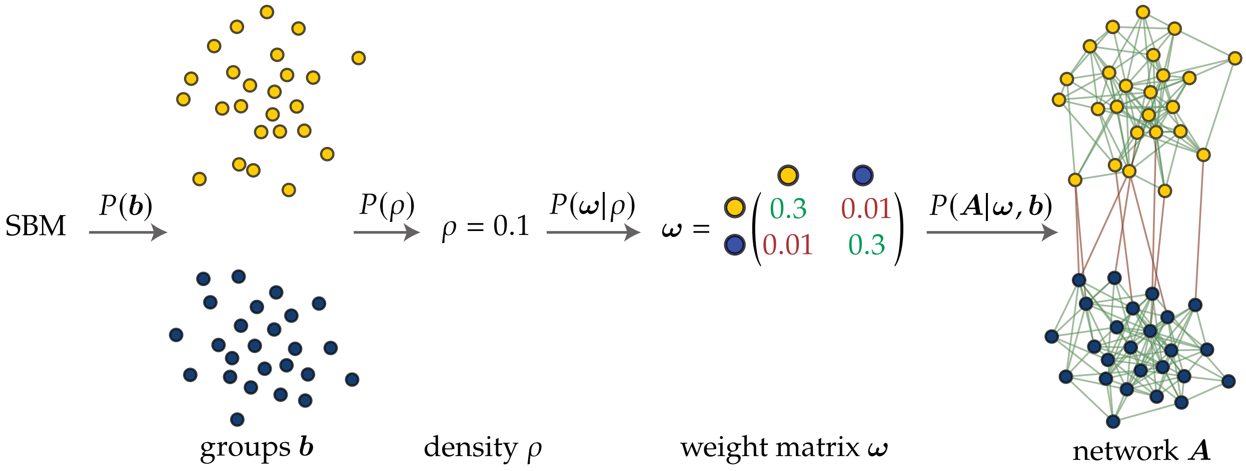

SBM
$P(\boldsymbol{b})$
$P(\rho)$
$\rho = 0.1$
$P(\boldsymbol{\omega}|\rho)$
$\boldsymbol{\omega} = \begin{pmatrix} 0.3 & 0.01 \\ 0.01 & 0.3 \end{pmatrix}$
$P(\boldsymbol{A}|\boldsymbol{\omega}, \boldsymbol{b})$
groups $\boldsymbol{b}$
density ρ
weight matrix $\boldsymbol{\omega}$
network $\boldsymbol{A}$

Figure 1.12: Example generative process for the stochastic block model of group structure. The group each node belongs to $\boldsymbol{b}$, represented by the colors, is first sampled. The overall density ρ is then sampled and used to generate the symmetric weight matrix ω. In the final network $\boldsymbol{A}$ the density of edges between a node in group r and a node in group s is given by the entry ω_{rs}. In this example the edges within groups (colored green) are much more likely than edges between groups (red), resulting in an assortative group structure.

by group. In this example the weight matrix generates an assortative group structure, although the choice of $\boldsymbol{\omega}$ can specify an arbitrary pattern of connections.

In many applications, we will mainly be interested in inferring the group structure $\boldsymbol{b}$ of the network rather than the weight matrix. For this purpose we *marginalize* over the possible weight matrices $\boldsymbol{\omega}$ to obtain the integrated likelihood

$$\begin{aligned} P(\boldsymbol{A}|\boldsymbol{b}, \rho) &= \int P(\boldsymbol{A}|\boldsymbol{\omega}, \boldsymbol{b}) P(\boldsymbol{\omega}|\rho) d\boldsymbol{\omega} \\ &= \underbrace{\frac{\prod_{r<s} M_{rs}! \prod_r M_{rr}!!/n_r^{m_r}}{\prod_{i<j} A_{ij}! \prod_i A_{ii}!!}}_{\text{multinomial}} \underbrace{\prod_{r<s} \frac{(\rho n_r n_s)^{M_{rs}}}{(\rho n_r n_s + 1)^{M_{rs}+1}} \prod_r \frac{(\rho n_r^2/2)^{M_{rr}/2}}{(\rho n_r^2/2 + 1)^{M_{rr}/2+1}}}_{\text{geometric}} \\ &= P(\boldsymbol{A}|\boldsymbol{M}, \boldsymbol{b}) P(\boldsymbol{M}|\boldsymbol{n}, \rho). \end{aligned} \tag{1.50}$$

This expression again factorizes into a microcanonical picture. The edge count matrix entries are distributed geometrically, while the positions of the edges between groups are distributed multinomially.

In terms of this integrated likelihood, we can use Bayes' law to write the posterior distribution over potential group structures and densities

$$P(\boldsymbol{b}, \rho | \boldsymbol{A}) = \frac{P(\boldsymbol{A}|\boldsymbol{b}, \rho)P(\boldsymbol{b})P(\rho)}{P(\boldsymbol{A})} \tag{1.51}$$

This can be a complex multi-modal distribution as many group structures are possible, although often only the maximum *a posteriori* (MAP) estimate of the community structure is reported as the best fit. For a more complete picture we can sample potential group structures from this posterior distribution using Markov Chain Monte Carlo (MCMC) methods detailed in Appendix B.6. These sampled partitions can then be summarized in a number of ways [76, 72], including the consensus clustering method discussed in Appendix B.7.

In Figure 1.13 we show the results of this posterior sampling for the Division IA and Big Ten networks from earlier this section. We plot the posterior distributions of the number of found communities q for each network. Since a single community $q = 1$ reduces to the Erdős-Rényi random graph, its density in the posterior distribution reflects the relative evidence of the Erdős-Rényi graph and the full SBM. The posteriors thus show that the Big Ten conference is best described by the Erdős-Rényi model while the full Division IA network warrants the stochastic block model. From the peak of the posterior, we also observe that the SBM finds $q = 10$ communities in the Division IA network, again slightly less than the 12 "true" conferences. These model selections are also confirmed by the cross-validation performances of the models on the data sets reported in Table 1.3.

Data set	Division IA		Big Ten conference	
Model	Erdős-Rényi	SBM	Erdős-Rényi	SBM
$\hat{q}_{\text{MAP}}$	N/A	10	N/A	1
$H(\boldsymbol{A})$	4335.0	**3584.5**	**171.8**	172.8
$H(\boldsymbol{A}^{\text{test}}\|\boldsymbol{A}^{\text{train}})$	859.0	**710.5**	**33.89**	34.03

Table 1.3: Table of the number of communities $\hat{q}$ found by the SBM, the Bayesian evidence $H(\boldsymbol{A})$, and posterior-predictive $H(\boldsymbol{A}^{\text{test}}|\boldsymbol{A}^{\text{train}})$ for Erdős-Rényi and stochastic block models on the Division IA and Big Ten conference football match networks.

Although the stochastic block model thus offers a Bayesian framework for inferring and justifying community structures, the model still has certain limitations. For one, the model shares the same homogeneous distribution of degrees as the Erdős-Rényi random model it is based upon. In order to create a model that captures both the group structure of the SBM and the degree variation of the configuration model we will need to *degree-correct* the SBM as described in Section 2.1 [69]. There we will observe that different networks

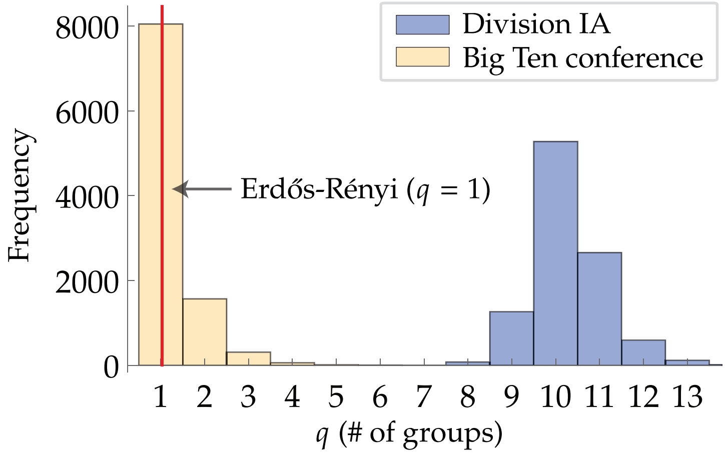


Figure 1.13: Posterior distributions of the number of groups q found by the stochastic block model within the networks of football matches within only the Big Ten conference (orange) and the entire Division IA (blue). The special case of $q = 1$, for which the SBM reduced to the Erdős-Rényi random graph is highlighted in red. We observe that while there is considerable evidence of group structure in the Division IA network, likely into 10 groups. The Big Ten conference, however, does not meaningfully have internal group structure.

call for various amounts of this degree correction, just as the general configuration model encompasses a spectrum of degree variation.

Secondly, the model suffers from a so-called *resolution limit* where the model is unable to find communities smaller than a certain size, even when they are well-separated [53]. This drawback is shared with modularity maximization, as seen by both the SBM and modularity maximization finding fewer than the true $q = 12$ communities of the football network. We discuss why this occurs and how to adjust the SBM to address this effect in Section 2.2.

More fundamentally, all stochastic block models, including the novel ones presented in this thesis cannot find communities beyond a *detectability threshold* where the assortative (or disassortative) preference that defines the groups is too weak to recover among the noise [6]. In fact, this threshold is an inherent limitation of any method for identifying such groups. In Section 3.1 we will observe how a wide variety of community detection algorithms run into the same barrier. This threshold is analogous to the phase transition of the Ising model foundational to statistical physics, a perspective we discuss more in Appendix A.5 [92]. Although some sufficiently weak group structures can not be reliably inferred, this thesis introduces methods to uncover groups closer to this limit in Chapter 2.

1.2.4 Bradley-Terry model

In the last section we discussed how the stochastic block model can be used to understand group structures in networks. In this section we describe how the Bradley-Terry model offers a similar perspective on hierarchies, such as those described in Section 1.1.3. By modeling these hierarchies we will be able to infer the rankings of the participants and predict the outcomes of unobserved matches. In this section we will also introduce a novel generalization of this model that allows us to directly infer the strength of these hierarchies.

Before describing the full Bradley-Terry model, we can first define hierarchies in terms of an objective function, just as the modularity intuitively describes the quality of a group structure. Many observers of such systems, especially in sports, may come to their own, often quite different conclusions about the appropriate ranking of the competitors. Out of these possibilities, it is useful to have a consistent benchmark for what constitutes a "good" or "fair" ranking given the results.

A natural starting point is to ask that a ranking is consistent with the observations in the following sense: the favorite, the higher ranked player, should typically win the match. To analyze a ranking in this way, we first specify it as vector $\boldsymbol{s}$ where player i is assigned score s_i, and player i is considered better than player j in the ranking if they have a higher score, $s_i > s_j$. In this notation, we can then count up the number of times that the favorite indeed wins as

$$m_{\text{win}}(\boldsymbol{A}, \boldsymbol{s}) = \sum_{ij} A_{ij} \mathbf{1}\{s_i < s_j\}. \tag{1.52}$$

We note that the adjacency matrix $\boldsymbol{A}$ is asymmetric in this definition since we are now describing a directed network.

In lieu of a known rating of the participants, we can then define a ranking by maximizing the number of times the favorite indeed wins as

$$\boldsymbol{s}* = \text{argmax}_{\boldsymbol{s}} m_{\text{win}}(\boldsymbol{A}, \boldsymbol{s}). \tag{1.53}$$

This strategy is known as *minimum violation ranking* since it is equivalent to minimizing the number of upset wins, or "violations" of the ranking. It is computationally demanding to find the true optimal ranking in this sense, and heuristic algorithms are often employed [27].

We can also use the number of favorite wins m_{win} as a test statistic to establish the presence of a hierarchy in the same manner as the modularity demonstrates group structure.

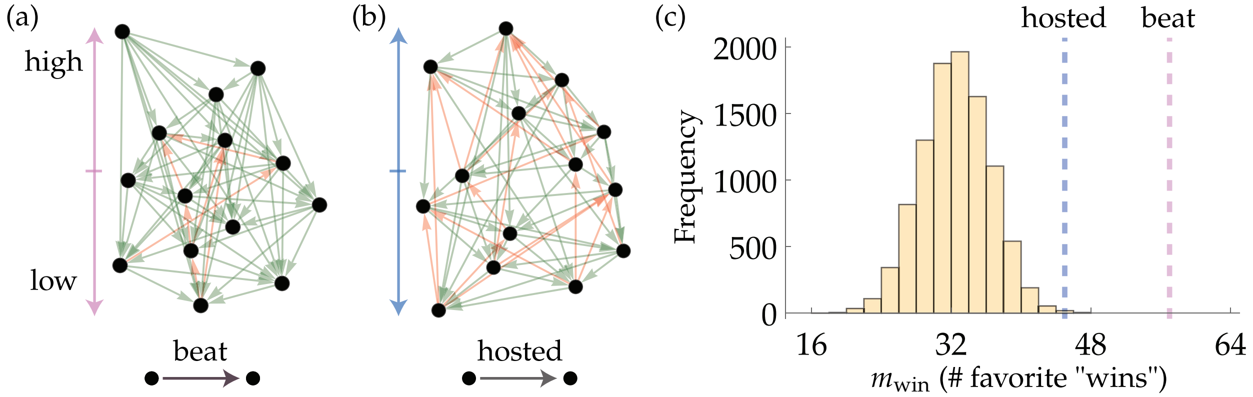


Figure 1.14: College football matches played in the 2022 Big Ten conference football season. (a) Arrows represent that one team beat another, teams are vertically arranged to minimize the number of upset victories (in red). (b) Arrows indicate that one team hosts another. Teams are vertically positioned to minimize ranking violations. (c) The number of times the favorite "wins" in each context, beating or hosting, are compared against 10,000 simulations where outcomes are sampled as fair coin flips.

As a null hypothesis, we use a "fair coin" model that either team is equally likely to prevail in any given match regardless of their ranking. As an example, we can ask whether there is significant evidence of a hierarchy present in the pattern of football victories in the 2022 season Big Ten conference [2], or if the outcomes could be just as well explained by coin flips. Figure 1.14a plots this network with arrows pointing from the winner to the loser of each match and the 14 teams vertically arranged according to the minimum violation ranking. In this hierarchy, $m_{\text{win}} = 57$ of the $m = 64$ total matches are won by the favored team (highlighted in green).

In Figure 1.14c this mark is compared against we could expect if each match was pure chance. In the fair coin model, the favorite wins half the time for an average of $m_{\text{win}} = 32$. However due to random fluctuations the favorites may happen to accumulate more wins than expected. Across the 10,000 plotted samples, however, in no simulation did the favorite win more than $m_{\text{win}} = 47$ matches. Since this is far from the number observed within the football hierarchy, we can reject the fair coin model and conclude that genuine gaps in skill are driving the match outcomes.

In Figure 1.14b we plot another network of the very same matches within the conference, but now where a "win" indicates that one team hosted the other rather than beat them. Intuitively we might expect this interaction to not be strongly driven by a hierarchy among the teams, but rather be more akin to the coin flip model. Yet, if again identify the minimum violations ranking, we can construct an ordering of the teams where the favorite wins $m_{\text{win}} = 45$ times. Returning to the significance testing, in only 8 of the 10,000

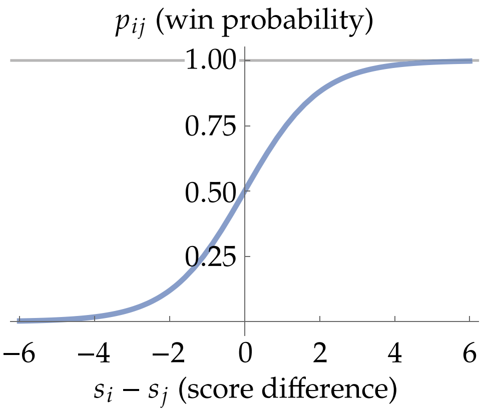


Figure 1.15: Win probability p_{ij} as a function of score difference $s_i - s_j$ in the Bradley-Terry model as in Eq. (1.54).

simulations did random flips generate more favorite wins than this mark for a p-value $P < 0.001$. In isolation this would be seen as very strong evidence of our found hierarchy, although since this ranking is chosen among $14! = 87178291200$ possible orderings, the low p-value may not be as impressive as it first appears.

In order to establish that a hierarchy is present in a more intrinsic manner, and to make predictions (e.g. who will win the next game?) we will again turn to generative models. Particularly, we consider the Bradley-Terry model, named after the work of R. Bradley and M. Terry who described it in 1952 [20], although it was (unknown to them) first introduced much earlier, by Zermelo in 1929 [143]. In the model, the probability p_{ij} that node i beats node j is assumed to depend upon the difference $s_i - s_j$ between their scores. Specifically, the win probability is taken to be a logistic sigmoid function of this difference as

$$p_{ij} = \frac{1}{1 + e^{-(s_i - s_j)}}. \tag{1.54}$$

This function, plotted in Figure 1.15, has a number of intuitive properties. If the two participants are evenly matched, $s_i = s_j$, $p_{ij} = 1/2$ and each competitor is equally likely to prevail. If node i is considerably better than node j, $s_i \gg s_j$, they are very likely to win as $p_{ij} \to 1$. Conversely if they are thoroughly outmatched and $s_i \ll s_j$, node i is unlikely to win as $p_{ij} \to 0$. This *score function* also critically assigns a behavioral meaning to a particular score differential. If node i is one "unit" above node j in the hierarchy, they will win with probability

$$p_{ij} = \frac{1}{1 + e^{-1}} \approx 0.731. \tag{1.55}$$

This gap then defines the meaning of a "tier" or "level" of skill differential, a difference that leads the favorite to win roughly 70-75% of the time.

Compiling these win probabilities across all of the matches then yields the Bradley-Terry model likelihood

$$P(\boldsymbol{A}|\boldsymbol{s}) = \prod_{ij} p_{ij}^{A_{ij}} = \prod_{ij} \left(\frac{1}{1 + e^{-(s_i - s_j)}} \right)^{A_{ij}}. \tag{1.56}$$

Most treatments of this model directly infer the maximum-likelihood values of the scores, although this approach can be prone to overfitting. In this thesis we instead consider a Bayesian treatment of the Bradley-Terry model, and find that this perspective gives insight to the nature of the hierarchies considered.

Particularly, on each score s_i we introduce an independent Gaussian prior of width $\beta/\sqrt{2}$ for parameter $\beta > 0$,

$$P(\boldsymbol{s}|\beta) = \prod_{i=1}^{n} \frac{1}{\beta\sqrt{\pi}} e^{-s^2/\beta^2}. \tag{1.57}$$

This choice is made so that the distribution of the *differences* in scores $s_i - s_j$ follow a Gaussian distribution of width β. The β parameter therefore controls the typical difference in score between two random nodes. Given the meaning of one unit of score difference, β counts how many layers of skill or status are present in the hierarchy between the typical pair. With this interpretation, we define this parameter in Chapter 4 as the *depth of competition* [67].

Starting with a half-Cauchy prior Eq. (4.15) over the depth, Figure 1.16 demonstrates the full generative process in the Bradley-Terry model. Figure 1.16a shows an example where the depth $\beta = 3$ is relatively high. Since the typical difference between scores is large, most matches are won by the favorite. In contrast, Figure 1.16b illustrates a lower depth of $\beta = 1$ where the now smaller score differences lead to more upsets – violations of the ranking. From this perspective we also notice that if the depth is $\beta = 0$ all of the scores must be the same and so all matches are even, recovering the fair coin null model considered earlier.

Given a network $\boldsymbol{A}$ we can then take MCMC samples from the posterior distribution of the scores $\boldsymbol{s}$ and the depth β,

$$P(\boldsymbol{s}, \beta|\boldsymbol{A}) = \frac{P(\boldsymbol{A}|\boldsymbol{s})P(\boldsymbol{s}|\beta)P(\beta)}{P(\boldsymbol{A})}. \tag{1.58}$$

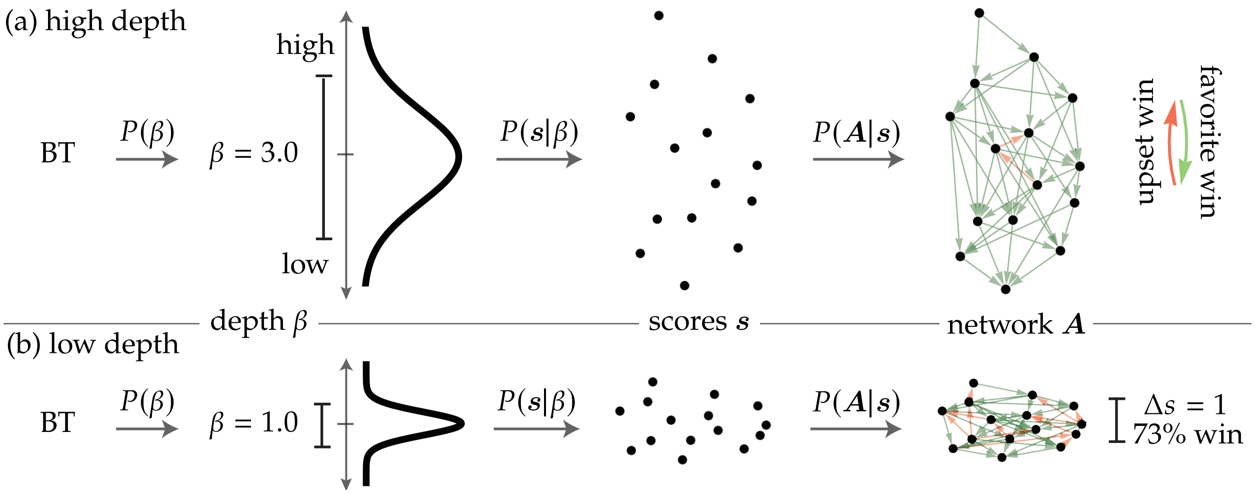


Figure 1.16: Example generative process for the Bradley-Terry model of hierarchies. (a) The depth β is first fixed, then the scores $\boldsymbol{s}$ are sampled from a Gaussian of that width, represented by the vertical distribution with width $\beta = 3$. The differences of these scores then inform who wins each match, where the higher score participant is favored. (b) This same process starting with a depth of $\beta = 1$. This smaller depth leads to smaller differences in scores and so more upset wins, colored in red.

In Figure 1.17 the resulting posterior distributions of the depth for the football victories and football hosting data sets are plotted. For the pattern of victories, we can infer a depth of $\hat{\beta}_{\text{MAP}} \approx 1.9$ within the conference, indicating that there are roughly 2 levels of play between a random pair of teams. Particularly, the victories posterior clearly excludes the case $\beta = 0$, indicating strong evidence for the hierarchy over the fair coin model. On the other hand, the pattern of hosting among the teams favors a depth of 0, indicating that hierarchical structure is not present and that the fair coin model is more appropriate. These conclusions are supported by direct computations of the description length and predictive power in Table 1.4.

Data set	Football wins		Football hosting	
Model	Fair coin	B-T	Fair coin	B-T
$\hat{\beta}_{\text{MAP}}$	N/A	1.92	N/A	0
$H(\boldsymbol{A})$	64.00	**54.43**	**64.00**	65.92
$H(\boldsymbol{A}^{\text{test}}\vert\boldsymbol{A}^{\text{train}})$	12.93	**10.88**	**12.93**	13.06

Table 1.4: Table of best the measured depth $\hat{\beta}_{\text{MAP}}$, Bayesian evidence $H(\boldsymbol{A})$, and posterior-predictive $H(\boldsymbol{A}^{\text{test}}|\boldsymbol{A}^{\text{train}})$ for the fair coin and Bradley-Terry (B-T) models on the football wins and hosting data sets.

In Chapter 4, we discuss how this model can be further generalized to model a "luck"

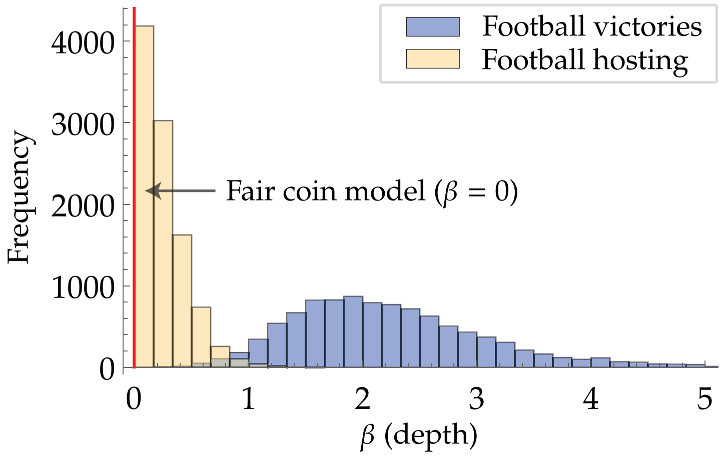


Figure 1.17: Posterior distribution of the depth β inferred by the general Bradley-Terry in the football victories and football hosting data sets. There is strong evidence of a hierarchical structure among the pattern of victories, while the pattern of hosting does not exclude the possibility that outcomes are fully random, highlighted at $\beta = 0$.

component of hierarchies where upsets are possible even between competitors of very different status. This generalization also includes the minimum violation ranking originally considered in this section, allowing for a direct comparison of the two ranking methods. In that chapter we also infer the depths of a variety of different data sets, ranging from sports and games to human and animal social hierarchies. Figure 1.18 summarizes these results.

The sports and games we consider in Figure 1.18 – highlighted in red – vary in their depth yet tend to have a low β compared to other contexts. Human social hierarchies (in green), such as patterns of friendships or of faculty hiring between universities, have a deeper, steeper hierarchy than these sports. And more than both these cases, we find that the animal social hierarchies (in blue) are very strict, with exceedingly large β values in some cases. Although the model does not know the source of a given network, whether it represents sports matches, human or animal interactions, it consistently categories the examples by measuring the depth of each context. With this ability to measure properties like the inequality in a hierarchy in hand, we can start to speculate and investigate what factors lead to these contrasting effects. For example, sports leagues are designed to be competitive for entertainment value, whereas social hierarchies are subject to no such incentives. By considering all of these applications within a unified model we can draw comparisons between them.

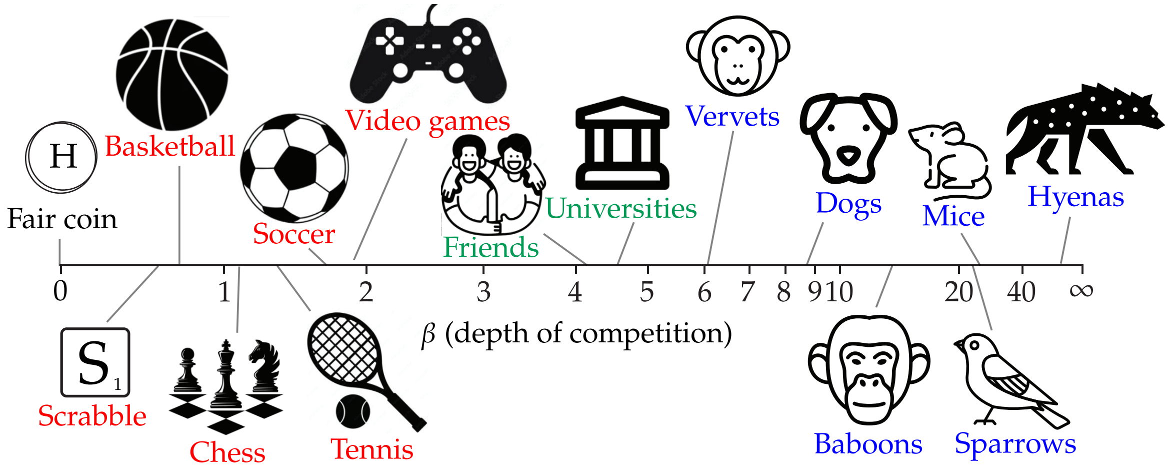


Figure 1.18: Depths of hierarchies inferred by our model across various applications. A depth of $\beta = 0$ corresponds to outcomes determined by fair coin flips while higher values of β indicate deeper, stricter hierarchies. Full posterior distributions of β for these cases are given in Figure 4.2.

1.3 Contributions

This chapter has reviewed the basic definitions, frameworks, and models that the main contributions of this thesis build upon. In the following chapters we will explore these models in greater detail and demonstrate how they can be leveraged to gain insight to a wide array of empirical settings.

In Chapter 2, we propose extensions of the simple stochastic block model discussed in Section 1.2.3. In Section 2.1 we first discuss the degree-correction of the stochastic block model, and how the model can be used to infer the extent of degree inequality within groups. In Section 2.2 we describe how the SBM can also be extended in order to directly measure the assortativity of a given group structure and so overcome the resolution limit that typically prevents the model from detecting small communities.

In Chapter 3, we consider refinements to the mutual information widely used to quantify the similarity of two partitions of the same set objects, including the outputs of community detection algorithms. We address two sources of bias of the usual measure. First, a typically neglected term that biases the measure towards labelings with many groups, and second a biased normalization of the mutual information. After making these corrections, we use the measure to demonstrate the relative performance and limitations of a variety of community detection methods and demonstrate how our conclusions depend on the form of the measure.

In Chapter 4, we discuss extensions of the Bradley-Terry model, and infer not only the depth of a hierarchy but also its inherent "luck." In this generalization we can also directly compare the Bradley-Terry model against the minimum violations ranking described in Section 1.2.4. We use this extension to compare a variety of hierarchies and demonstrate that our new model can better predict unobserved match outcomes.

In Chapter 5 we conclude with a summary of the work presented here and possible future directions of inquiry.

Chapter 2

Group Structure

In this chapter, we explore generalizations of the stochastic block model (SBM) described in Section 1.2.3, to model and understand a wider variety of group structures in networks. These adjustments address inequality of degrees within groups, overall assortative preference, and variation in group relationships. In real networks we identify statistically significant evidence of these features and improve performance of the SBM. These refinements give a more detailed picture of how latent groups inform network structure across applications.

2.1 Degree-correction of the stochastic block model

2.1.1 Introduction

In this section, we first consider an extension of the traditional SBM that incorporates the variation in node degrees found in the configuration model. Known as *degree-correction* [69], this adjustment makes the model more permissive of nodes with varying degrees within the same group.

The usual, non degree-corrected SBM typically groups nodes of similar degree together. This behavior aligns with the model's foundational assumption that connection probabilities are informed only by group identities, and thus that nodes within each group behave identically, as in an Erdős-Rényi random graph. If network nodes can only be differentiated by group, high degree nodes should belong to different groups than low degree nodes since the two categories behave differently.

In some applications, however, this tendency identifies groups that differ from understood divisions. In a social networks, for example, individuals typically interact more often with others who share certain group identities, like race or gender [91]. Within these groups, significant variation in degrees can occur, as some individuals within a group are more prone to making connections than others. The traditional SBM would sooner classify these individuals based upon the number of connections they make than upon the bias of those connections, which may be more relevant to the "true" group identities. A more flexible model is needed to identify such groups that contain individuals with a diversity of degrees.

Several SBM extensions have been introduced to capture increased degree heterogeneity [132, 146, 107]. In this work we focus on the simple model proposed by Karrer and Newman [69], commonly referred to as the degree-corrected SBM. In this approach the interior of each group is governed by the configuration model rather than the non degree-corrected Erdős-Rényi model. The degree-corrected SBM thus inherits the greater degree variation exhibited by the configuration model and identifies groups with broader degree distributions than the typical model.

This alternative presents a natural model selection problem between the degree-corrected and non degree-corrected SBMs. By construction the two models can identify very different group structures of the same network. Although each inferred group structure is valid within the assumptions of its respective model, we adjudicate between them by assessing which model does a holistically better job at reproducing the network. Prior work has explored this question through approximate belief propagation methods to compare the Bayesian evidence of the two models [139].

Our approach introduces a *general degree-corrected SBM* that embeds both the non degree-corrected and traditionally degree-corrected SBMs as special cases. This is accomplished by building the SBM based upon the general configuration model introduced in Section 1.2.2, which generalizes both the Erdős-Rényi and usual configuration models. Just as the general configuration model can then assess the relative evidence of the models it contains, this general degree-corrected model allows for the direct comparison of the two SBMs without approximating their performances.

As in our earlier comparison of the configuration and Erdős-Rényi models, some real networks require the extra flexibility degree-correction affords while others do not. In our framework we can infer not only when this generalization is appropriate but also measure the relevant degree variation within the groups, which we call the *in-group degree inequality* G_{in}. This Gini coefficient of latent node weights parallels the overall degree inequality G measured by the general configuration model. Through these measurements, the general degree-corrected model offers a nuanced picture of the interplay between degrees and group identities within networks.

2.1.2 Traditional degree-correction

In this section, we review the traditional degree-corrected SBM (DC-SBM) [69] and demonstrate how its assumptions and inferences differ from those of the non degree-corrected model presented in Section 1.2.3.

The usual, non-degree corrected SBM likelihood assumes that connection probabilities depend only on group identities. Specifically, given a group partition $\boldsymbol{b}$ and symmetric weight matrix $\boldsymbol{\omega}$, the edge counts follow Poisson distributions with expectations

$$\mathbf{E}A_{ij} = \omega_{b_i b_j}, \tag{2.1}$$

for an overall network likelihood

$$P(\boldsymbol{A}|\boldsymbol{\omega}, \boldsymbol{b}) = \prod_{i<j} \frac{\omega_{b_i b_j}^{A_{ij}} e^{-\omega_{b_i b_j}}}{A_{ij}!} \prod_i \frac{(\omega_{b_i b_i}/2)^{A_{ii}/2} e^{-\omega_{b_i b_i}/2}}{(A_{ii}/2)!}. \tag{2.2}$$

In this model, the expected degree of a node i is constant within its group $r = b_i$, equal to

$$\mathbf{E}k_i = \sum_{j=1}^{n} \mathbf{E}A_{ij} = \sum_{s=1}^{q} n_s \omega_{rs}. \tag{2.3}$$

While fluctuations in the degrees about this expectation are possible, they tend to be small, just as the degrees concentrate in the Erdős-Rényi model.

To induce greater degree variability we introduce to each node i a *node weight* $\theta_i > 0$ that represents the node's overall propensity to interact. Like in the configuration model, these node weights adjust the expected edge counts as

$$\mathbf{E}A_{ij} = \theta_i \theta_j \omega_{b_i b_j}. \tag{2.4}$$

Since these node weights appear multiplied by the weight matrix $\boldsymbol{\omega}$, their overall normalization is arbitrary as it can be absorbed by changes in the weight matrix. We thus normalize the weights to have an average value of 1 within each group as in [116],

$$\frac{1}{n_r} \sum_{i \in r} \theta_i = 1. \tag{2.5}$$

This choice is the natural generalization of the normalization Eq. (1.20) used in the configuration model. With this adjustment, the expected degrees of nodes in group r now vary with their node weights as

$$\mathbf{E}k_i = \sum_{j=1}^{n} \mathbf{E}A_{ij} = \theta_i \sum_{s=1}^{q} n_s \omega_{rs}. \tag{2.6}$$

By introducing variation in the node weights we can thus generate networks whose degrees vary more within groups than in the non degree-corrected SBM.

Incorporating these weights, the degree-corrected model likelihood is

$$\begin{aligned} P(\boldsymbol{A}|\boldsymbol{\theta},\boldsymbol{\omega},\boldsymbol{b}) &= \prod_{i<j} \frac{(\theta_i\theta_j\omega_{b_ib_j})^{A_{ij}} e^{-\theta_i\theta_j\omega_{b_ib_j}}}{A_{ij}!} \prod_{i} \frac{(\theta_i^2\omega_{b_ib_i}/2)^{A_{ii}/2} e^{-\theta_i^2\omega_{b_ib_i}/2}}{(A_{ii}/2)!} \\ &= \frac{\prod_i \theta_i^{k_i}}{\prod_{i<j} A_{ij}! \prod_i A_{ii}!!} \prod_{r<s} \omega_{rs}^{M_{rs}} e^{-n_r n_s \omega_{rs}} \prod_{r} (\omega_{rr}/2)^{M_{rr}/2} e^{-n_r^2 \omega_{rr}/2}. \end{aligned} \tag{2.7}$$

The models we discuss in this chapter will all share this likelihood and differ only in their priors on the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\omega}$. Since the number of these parameters grows with the system size, such differences in their prior distribution dramatically influence model behavior.

In this framework, the non degree-corrected SBM uses the Dirac-delta[1] prior

$$P(\boldsymbol{\theta}) = \prod_{i=1}^{n} \delta(\theta_i - 1), \tag{2.8}$$

which fixes the node weights at $\theta_i = 1$ and so recovers the non degree-corrected SBM likelihood Eq. (2.2). To define the traditional degree-corrected SBM, we instead use a uniform, maximum entropy prior over possible node weights $\boldsymbol{\theta}$. In terms of the volume of the product of q simplexes that respects the normalization condition Eq. (2.5), this is

$$P(\boldsymbol{\theta}|\boldsymbol{n}) = \prod_{r=1}^{q} \frac{(n_r - 1)!}{n_r^{n_r-1}}. \tag{2.9}$$

This prior depends on the group sizes $\boldsymbol{n}$ since they influence how many normalized choices of $\boldsymbol{\theta}$ can be made.

For both models we adopt independent and identical exponential priors on the entries

[1] Here we define the Dirac-delta function to correspond to point evaluation when integrated over the product of simplexes. The typical normalization over $\mathbf{R}^n$ would include a prefactor.

of the symmetric weight matrix $\boldsymbol{\omega}$ with expected density $\rho > 0$,

$$P(\boldsymbol{\omega}|\rho) = \prod_{r \leq s} \frac{1}{\rho} e^{-\omega_{rs}/\rho}, \tag{2.10}$$

along with an exponential prior on the density itself,

$$P(\rho) = e^{-\rho}. \tag{2.11}$$

In Section 2.2 we will observe how model behavior changes based on the form of this weight matrix prior. In this section, however, we will focus only on the influence of the node weight prior $P(\boldsymbol{\theta})$ to degree-correct the model.

By integrating both the node weights $\boldsymbol{\theta}$ and weight matrix $\boldsymbol{\omega}$ against these priors, we obtain the integrated, degree-corrected (DC) likelihood

$$\begin{aligned} P_{\text{DC}}(\boldsymbol{A}|\boldsymbol{b}, \rho) &= \int P(\boldsymbol{A}|\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{b}) P(\boldsymbol{\theta}|\boldsymbol{n}) P(\boldsymbol{\omega}|\rho) d\boldsymbol{\theta} d\boldsymbol{\omega} \\ &= \underbrace{\frac{\prod_i k_i! \prod_{r<s} M_{rs}! \prod_r M_{rr}!!}{\prod_{i<j} A_{ij}! \prod_i A_{ii}!! \prod_r m_r!}}_{\text{stub-matching}} \underbrace{\prod_r \binom{m_r + n_r - 1}{n_r - 1}^{-1}}_{\text{uniform}} \\ &\quad \times \underbrace{\prod_{r<s} \frac{(\rho n_r n_s)^{M_{rs}}}{(\rho n_r n_s + 1)^{M_{rs}+1}} \prod_r \frac{(\rho n_r^2/2)^{M_{rr}/2}}{(\rho n_r^2/2 + 1)^{M_{rr}/2+1}}}_{\text{geometric}} \\ &= P(\boldsymbol{A}|\boldsymbol{k}, \boldsymbol{M}, \boldsymbol{b}) P_{\text{Unif}}(\boldsymbol{k}|\boldsymbol{m}, \boldsymbol{b}) P(\boldsymbol{M}|\boldsymbol{n}, \rho). \end{aligned} \tag{2.12}$$

By factorizing the likelihood, we highlight its microcanonical interpretation. From this perspective the generation process begins with the geometric distribution of the edge count matrix $\boldsymbol{M}$. The row (or column) sums $\boldsymbol{m}$ of this matrix then set the total degree within each group. In each group r, the assignment of the total degree m_r among the n_r nodes is then sampled uniformly among all possible assignments in $P_{\text{Unif}}(\boldsymbol{k}|\boldsymbol{m}, \boldsymbol{b})$. Note that this does not mean the resulting distribution of the individual degrees k_i is uniform. Rather, the full degree sequence $\boldsymbol{k}$ itself is selected uniformly among its possible configurations. Finally the network $\boldsymbol{A}$ is distributed according to the stub-matching likelihood $P(\boldsymbol{A}|\boldsymbol{k}, \boldsymbol{M}, \boldsymbol{b})$ among all possible networks that share the generated degree sequence $\boldsymbol{k}$ and edge counts $\boldsymbol{M}$ across the partition $\boldsymbol{b}$. This final stub-matching likelihood is a generalization of the stub-matching Eq. (1.28) used in the configuration model, now conditioned on the edge count matrix $\boldsymbol{M}$ across groups.

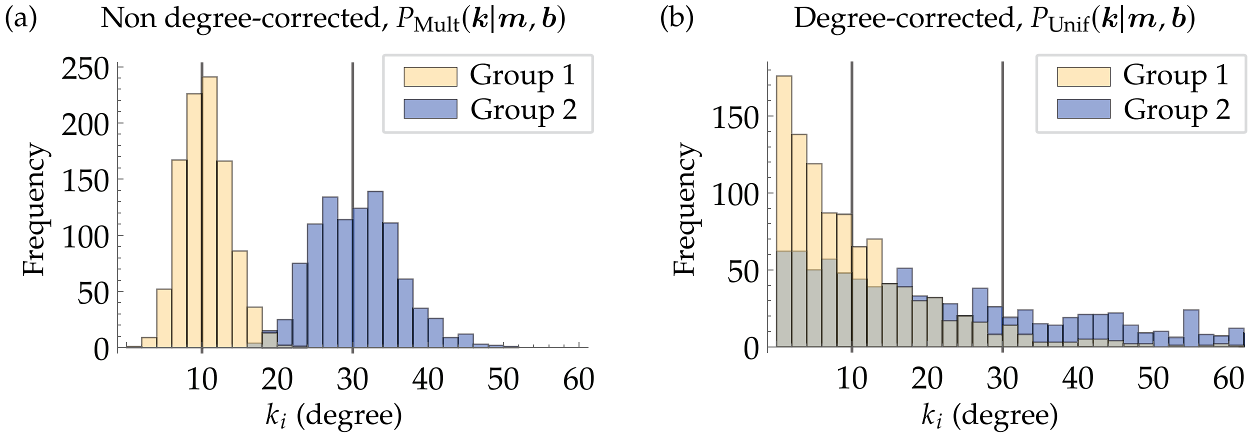


Figure 2.1: Examples of degree distributions $\boldsymbol{k}$ sampled from the prior distributions of the (a) non degree-corrected and (b) degree-corrected SBMs. These examples contain two groups of 1000 nodes each, group 1 (orange) has average degree 10, and group 2 (blue) has average degree 30. In the non degree-corrected case, the multinomially distributed degrees are concentrated about their group means and the two groups have little overlap in degree. In the degree-corrected case the degree distribution within each group is much broader, leading to greater overlap between the groups.

This microcanonical picture offers a direct view of how the degree-corrected SBM (DC-SBM) differs from the non degree-corrected SBM (NDC-SBM). The non degree-corrected likelihood, Eq. (2.2), can be factorized in the same manner as the degree-corrected likelihood as

$$P_{\mathrm{NDC}}(\boldsymbol{A}|\boldsymbol{b},\rho) = P(\boldsymbol{A}|\boldsymbol{k},\boldsymbol{M},\boldsymbol{b})P_{\mathrm{Mult}}(\boldsymbol{k}|\boldsymbol{m},\boldsymbol{b})P(\boldsymbol{M}|\boldsymbol{n},\rho), \tag{2.13}$$

only now with a multinomial distribution of the degrees within each group

$$P_{\mathrm{Mult}}(\boldsymbol{k}|\boldsymbol{m},\boldsymbol{b}) = \prod_r \frac{m_r!}{\prod_{i\in r} k_i! n_r^{k_i}}. \tag{2.14}$$

The degrees of the n_r nodes within group r are thus distributed as a symmetric multinomial, i.e. as though a fair n_r-sided die is rolled m_r times and the outcome counts determine the node degrees. As the total degree m_r within a group increases, this multinomial distribution concentrates around its expectation, an even distribution of the degrees $\mathbf{E}k_i = \frac{m_r}{n_r}$. Therefore, especially in large groups, the non degree-corrected SBM indeed has a prior expectation that nodes in the same group have similar degrees.

Figure 2.1 contains an example of this effect. In Figure 2.1a, the non degree-corrected SBM is used to generate the degree sequence of a network split into two groups of 1000 nodes each, where one group has average degree 10 and the other average degree 30. From the microcanonical formulation, this is equivalent to sampling $\boldsymbol{k}$ from the multinomial distribution $P_{\text{Mult}}(\boldsymbol{k}|\boldsymbol{m}, \boldsymbol{b})$ for a partition $\boldsymbol{b}$ of 2000 nodes into equally sized groups, and $\boldsymbol{m} = (10000, 30000)$. In the non degree-corrected case, we observe that the two groups have little overlap in their degree distribution as each group is concentrated about its mean degree. In contrast, the degree sequence $\boldsymbol{k}$ sampled from the degree-corrected $P_{\text{Unif}}(\boldsymbol{k}|\boldsymbol{m}, \boldsymbol{b})$ in Figure 2.1b has a much wider range of degrees within each group. The two groups are therefore no longer solely distinguished by degree as their distributions overlap[2].

This difference in behavior of the priors impacts the inferences drawn by each model. Since the non degree-corrected model has a prior expectation that nodes of the same group have similar degrees, it systematically places nodes of the same degree into the same group and nodes of different degrees into different groups.

This effect is borne out in Figure 2.2, which illustrates the consensus partitions inferred by these models in a network of the co-purchasing of political books near the 2004 U.S. presidential election [1]. In the figure the size of each node represents its degree, which covers a wide range as some books are purchased with many others, while others are rarely co-purchased. As discussed in Section 1.1.2, this network also shows signs of political polarization, and we plot the generally more conservative books near the top of the network, and more liberal books near the bottom.

Figure 2.2a depicts the groups of books inferred by the non degree-corrected SBM, represented by the colors of the nodes. The model roughly splits the network into groups along partisan lines, where the liberal books and the conservative books largely belong to their own groups. Within each of these factions we also observe that groups are further differentiated by degree. In the figure the blue group consists only of liberal books of high degree, and the purple group contains only liberal books with low degree. The inferred groups appear to be informed by a combination of both the partisan lean and the degree of each book.

In contrast, the degree-corrected model groups together all the books that have strongly liberal and conservative tilt into the red and blue groups of Figure 2.2b. Since the degree-corrected model is more permissive of mixed degrees within groups, the groups it finds have less to do with the degrees of the nodes and instead the relative biases in their co-purchasing.

[2] The marginal distribution of the degrees in the uniform distribution $P_{\text{Unif}}(\boldsymbol{k}|\boldsymbol{m}, \boldsymbol{b})$ also concentrates when the total degree is sufficiently large, although this occurs more slowly than in the multinomial distribution $P_{\text{Mult}}(\boldsymbol{k}|\boldsymbol{m}, \boldsymbol{b})$.

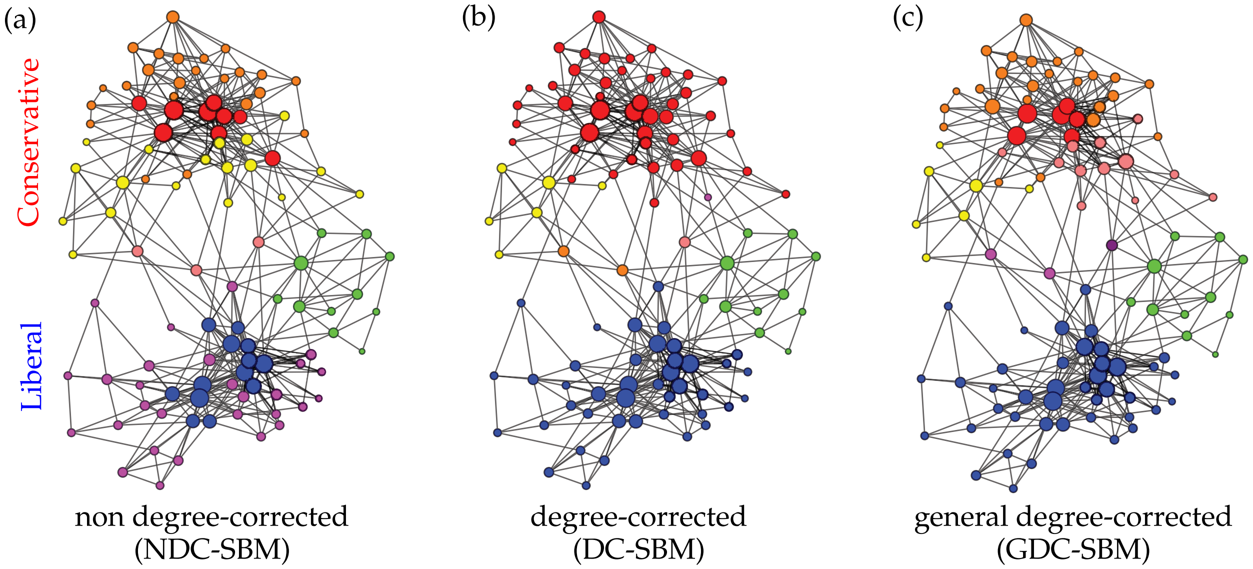


Figure 2.2: Inferred group structures of the political books network [1]. The size of each node represents its degree and the vertical position roughly indicates each book's political lean. The node colors represent groups found by each model. The (a) non degree-corrected SBM places high and low degree nodes into separate groups, the (b) degree-corrected version finds groups spanning many degrees, and the (c) general degree-corrected model has a mixed behavior.

The two models thus come to quite different conclusions about the underlying group structure of the network based upon their differing assumptions about the nature of the groups. In order to select between these options we will introduce a generalized model that interpolates between the two variants to check which model better describes the network.

2.1.3 General degree-correction

Just as the Erdős-Rényi and configuration models are included in the general configuration model, we can unify both the non degree-corrected and degree-corrected models of the previous section into a single *general degree-corrected SBM*.

The earlier models are distinguished only by their prior on the node weights $\boldsymbol{\theta}$. If we introduce a product of Dirichlet distributions with concentration parameter $\alpha_{\text{in}} > 0$,

$$P(\boldsymbol{\theta}|\boldsymbol{n}, \alpha_{\text{in}}) = \prod_{r=1}^{q} \frac{\Gamma(n_r\alpha_{\text{in}})}{n_r^{n_r-1}\Gamma(\alpha_{\text{in}})^{n_r}} \prod_{i\in r} \left(\frac{\theta_i}{n_r}\right)^{\alpha_{\text{in}}-1} \tag{2.15}$$

$$= \prod_{i=1}^{n} \theta_i^{\alpha_{\text{in}}-1} \prod_{r=1}^{q} \frac{\Gamma(n_r\alpha_{\text{in}})}{n_r^{n_r\alpha_{\text{in}}-1}\Gamma(\alpha_{\text{in}})^{n_r}}, \tag{2.16}$$

we can generalize both cases. Here $\alpha_{\text{in}} = 1$ corresponds to the uniform prior Eq. (1.24) and therefore the degree-corrected model. Likewise, the limit $\alpha_{\text{in}} \to \infty$ approaches the Dirac-delta prior Eq. (1.29) and so recovers the non degree-corrected model.

As α_{in} increases, the node weights $\boldsymbol{\theta}$ and therefore the degrees $\boldsymbol{k}$ become increasingly concentrated within each group. We thus refer to the parameter as the *in-group degree homogeneity*. To infer and interpret this parameter we can borrow techniques used for the overall degree homogeneity α of the general configuration model. To infer the parameter we adopt the same half-Cauchy prior distribution with median 1,

$$P(\alpha_{\text{in}}) = \frac{2}{\pi(\alpha_{\text{in}}^2 + 1)}. \tag{2.17}$$

We may also similarly transform the homogeneity parameter to reflect the inequality of the underlying node weights $\boldsymbol{\theta}$ within the groups as measured by the Gini coefficient

$$G_{\text{in}} = \binom{\alpha_{\text{in}} - 1/2}{\alpha_{\text{in}}}. \tag{2.18}$$

This *in-group degree inequality* is $G_{\text{in}} = 0$ for the non degree-corrected model where each node weight is identical, and $G_{\text{in}} = 0.5$ for the degree-corrected model where the weights vary. As the Gini G_{in} runs from 0 to 1, the general model can therefore realize levels of degree-correction both weaker and stronger than the typical degree-corrected model. Table 2.1 summarizes the choices of these parameters that correspond to these special cases.

Integrating the node weights against this generalized prior, the general degree-corrected SBM likelihood is

$$\begin{aligned} P(\boldsymbol{A}|\boldsymbol{b}, \rho, \alpha) &= \int P(\boldsymbol{A}|\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{b})P(\boldsymbol{\theta}|\boldsymbol{n}, \alpha)P(\boldsymbol{\omega}|\rho)d\boldsymbol{\theta}d\boldsymbol{\omega} \\ &= P(\boldsymbol{A}|\boldsymbol{k}, \boldsymbol{M}, \boldsymbol{b})P(\boldsymbol{k}|\boldsymbol{m}, \boldsymbol{b}, \alpha)P(\boldsymbol{M}|\boldsymbol{n}, \rho) \end{aligned} \tag{2.19}$$

where the distributions of the network $P(\boldsymbol{A}|\boldsymbol{k}, \boldsymbol{M}, \boldsymbol{b})$ and the edge counts $P(\boldsymbol{M}|\boldsymbol{n}, \rho)$

Model	α_{in}	G_{in}
Non degree-corrected SBM (NDC-SBM)	∞	0
Degree-corrected SBM (DC-SBM)	1	1/2
General degree-corrected SBM (GDC-SBM)	inferred	inferred

Table 2.1: Summary of the parameter values of the general degree-corrected SBM that correspond to the models it generalizes. For each model the in-group degree homogeneity α_{in} and corresponding in-group degree inequality G_{in} are given. The general model infers these parameters from the network.

are shared with Eq. (2.12), but the degree distribution is generalized to the product of Dirichlet-multinomial distributions

$$P(\boldsymbol{k}|\boldsymbol{m},\boldsymbol{b},\alpha) = \prod_{r=1}^{q} \binom{m_r + n_r\alpha - 1}{n_r\alpha - 1}^{-1} \prod_{i\in r} \binom{k_i + \alpha - 1}{\alpha - 1}. \tag{2.20}$$

The special cases $\alpha = 1$ and $\alpha \to \infty$ of this Dirichlet-multinomial distribution then correspond to the appropriate degree distributions $P_{\text{Unif}}(\boldsymbol{k}|\boldsymbol{m},\boldsymbol{b})$ and $P_{\text{Mult}}(\boldsymbol{k}|\boldsymbol{m},\boldsymbol{b})$ of the contained models.

When applying this general model to our example network of political books, the best fit value of $\alpha_{\text{in}} \approx 25$ (or $G_{\text{in}} \approx 0.08$) lies between the non degree-corrected and degree-corrected models. As such, the consensus partition of the books, plotted in Figure 2.2c, is a hybrid of the partitions found by the other two models. In this example the liberal leaning books are lumped into a single group as in the degree-corrected model while the conservative books are differentiated by degree like the non-degree corrected case.

2.1.4 Results

In this section, we discuss the results of the variously degree-corrected models on a selection of real networks spanning a range of sources and scales. Table 2.2 lists the 14 data sets we consider in this work arranged by their in-group degree inequality $\hat{G}_{\text{in}}$ as inferred by the general degree-corrected SBM.

Since the general model includes the other types of degree-correction as special cases, the posterior distribution of the interpolating parameter G_{in} reflects the relative evidence of the models as described in Section 1.2.2. For some of these networks, the posterior distribution is peaked at or near $G_{\text{in}} = 0$, the non degree-corrected case. Other data sets have in-group inequalities closer to $G_{\text{in}} = 0.5$, the traditional degree-corrected case.

To establish statistical significance, we can consider the error bars on G_{in} compared to

Data set	$\hat{G}_{\text{in}}$	n	m	Description	Ref.
Football	0.00	115	613	Division IA college football matches, 2000	[57]
Power grid	0.00	4941	6594	Power lines in the Western U.S. Power Grid	[135]
Karate	0.08	34	77	Zachary's karate club friendship network	[142]
Books	0.08	105	441	Co-purchasing of U.S. political books, 2004	[1]
Food web	0.09	128	2075	Carbon exchanges between Florida species	[129]
Friends	0.17	198	946	High-school friend nominations	[128]
Dolphins	0.21	62	159	Interactions among New Zealand dolphins	[84]
Neurons	0.25	297	2148	Neural connections of the *C. elegans* nematode	[137]
Proteins	0.28	1706	3191	Binding interactions of human proteins	[123]
Coauthors	0.30	1589	2742	Co-authorships among network scientists	[97]
Words	0.30	112	425	Subsequent words in Dickens' *David Copperfield*	[97]
Internet	0.34	3015	5539	Traffic among internet Autonomous Systems	[81]
E-mail	0.42	1133	5452	E-mails sent within Rovira i Virgili University	[60]
Blogs	0.60	1490	16718	Hyperlinks between U.S. political blogs, 2004	[7]

Table 2.2: Data sets in order of increasing in-group degree inequality $\hat{G}_{\text{in}}$. The number of nodes n, edges m, and a short description are given for each network.

the special parameter values, both plotted along the y axis of Figure 2.3. For some of these data sets, including the political books example, the posterior distribution meaningfully overlaps with the non degree corrected case. In none of the data sets we consider, however, is there significant overlap with traditional degree-correction. The data sets instead exhibit a range of in-group inequalities not particularly close to the special case $G_{\text{in}} = 0.5$. Considering these posterior distributions, we can also (approximately) exclude other approaches to degree-correcting SBMs, including the hierarchical prior commonly used for microcanonical models [107], as detailed in Appendix B.9.

Much of this variation of degrees within the groups is driven by the global variation of degrees in the network. We can quantify this overall inequality across all degrees using the general configuration model and the degree inequality Eq. (1.35), which we notate in this section as G_{total}. Figure 2.3 plots this total degree inequality along the x axis against the in-group degree inequality G_{in}.

These two measures are correlated over our examples, although we consistently observe less degree variation within groups than in the overall network, $G_{\text{in}} \leq G_{\text{total}}$. For example, the "Football" network analyzed in Section 1.2.2 has a very concentrated degree distribution and therefore total degree inequality $G_{\text{total}} = 0$. Within each of the found groups, these degrees remain homogeneous at $G_{\text{in}} = 0$, in part since there is little variation to begin with. In contrast the "Blogs" network has both large total degree variation $G_{\text{total}} \approx 0.68$

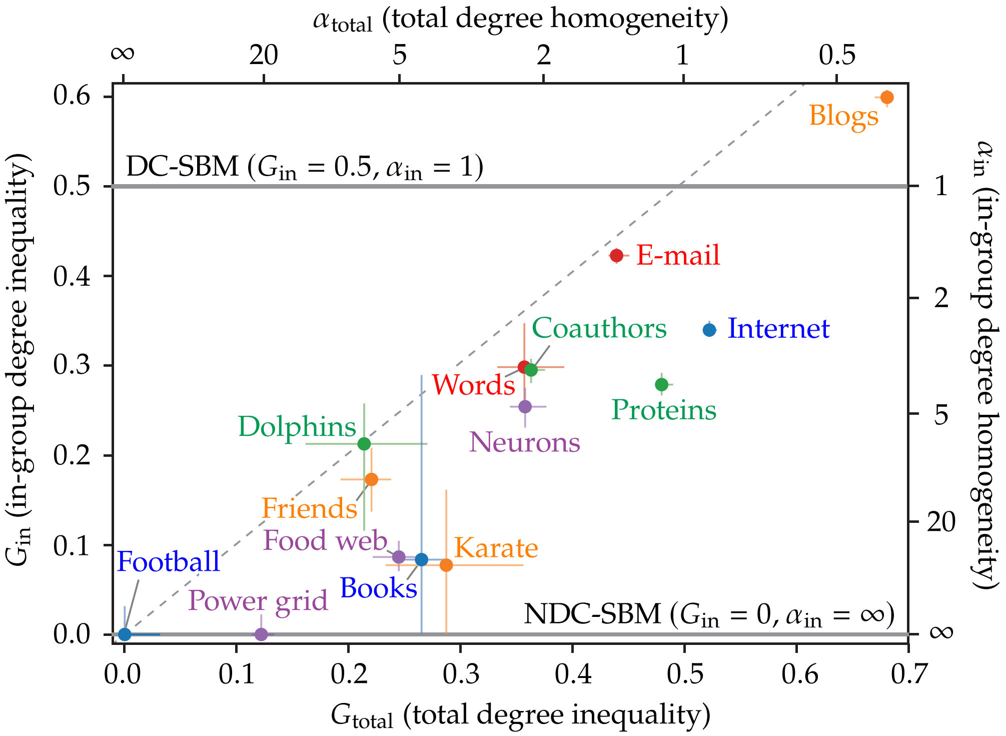


Figure 2.3: Total and within-group degree inequality as assessed by the configuration model and general degree-corrected SBM.

and in-group inequality $G_{\text{in}} \approx 0.60$.

Although the two measures are certainly coupled, the fraction of the overall degree inequality which manifests within the groups varies from network to network. While the "Karate" and "Words" examples have a similar level of overall variation in their degrees, the small G_{in} of the "Karate" data set indicates that this inequality mostly manifests between the groups, in contrast with the "Words" data set where the full variation occurs within the groups.

For another perspective on these models and data sets, we can consider their predictive power in cross-validation testing. In these tests, we randomly select 80% of our network to act as the training data set $\boldsymbol{A}^{\text{train}}$, then evaluate the posterior-predictive distribution of the remaining 20% portion of the network $\boldsymbol{A}^{\text{test}}$. At this ratio, where the test data set has an overall density $f = 0.25$ times that of the training data, this posterior-predictive is found by integrating over the model posterior distribution of all latent parameters as

$$P(\boldsymbol{A}^{\text{test}}|\boldsymbol{A}^{\text{train}}, f) = \sum_{\boldsymbol{b}} \int P(\boldsymbol{A}^{\text{test}}|\boldsymbol{\theta}, f\boldsymbol{\omega}, \boldsymbol{b}) P(\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{b}|\boldsymbol{A}^{\text{train}}) d\boldsymbol{\theta} d\boldsymbol{\omega}. \tag{2.21}$$

This can be accomplished through a combination of analytic integration and MCMC

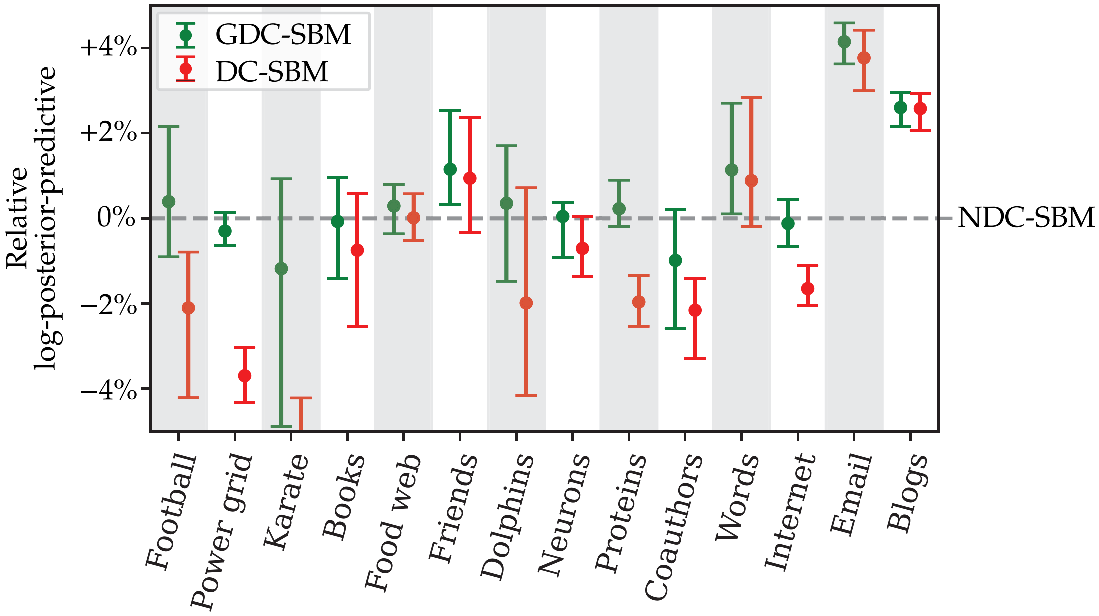


Figure 2.4: Figure of the log posterior-predictive performance of the degree-corrected models discussed in this work across 50 cross-validation splits of the models relative to the usual non degree-corrected SBM. Data sets are arranged in order of increasing degree inequality G_{in}.

posterior samples as detailed in Appendix B.8. This posterior-predictive then captures how effectively a model can extrapolate from one part of the network to another.

Figure 2.4 plots the log-posterior-predictive performances of the degree-corrected models normalized to the performance of the non degree-corrected model. The median and inter-quartile ranges across 50 cross-validation splits are recorded in each test. Although the posterior-predictive here quantifies the likelihood of recovering the entire withheld $\boldsymbol{A}^{\text{test}}$, suitably normalized these figures roughly give the improved probability of correctly predicting the position or existence of any given edge. Across these results we note that the general degree-corrected model performs best or within the cross-validation error of the best model for every data set. Although the traditional degree-corrected and non degree-corrected models perform well on certain networks, they do not match the comprehensive performance of the general model.

The data sets are presented in the figure in order of increasing in-group degree inequality G_{in} from left to right. From this arrangement we can observe that the non degree-corrected model performs better for low G_{in} networks while the degree-corrected model performs better for the high G_{in} examples. This conclusion aligns with our interpretation of the posterior distribution of G_{in} and its implications for the relative evidence of these models. As the general degree-corrected model can identify whether a network has low G_{in}, high G_{in}, or somewhere in between, it can make predictions accordingly and generally match the performance of the best model available in each case.

2.1.5 Conclusions

In this work we have described an extension of the stochastic block model that allows an arbitrary level of degree heterogeneity within groups. In doing this we directly generalized both the usual SBM and the traditionally degree-corrected SBM, allowing for their direct comparison. In tests on real networks, the general degree-corrected model outperforms existing SBMs by model evidence and predictive power metrics. The model also allows us to directly measure the extent to which overall inequality in the degree distribution manifests inside groups, and so probe the relationship between node degree and group identity. The networks we consider exhibit a wide range of degree inequalities, presenting an opportunity to understand what drives these differences in context.

From a modeling perspective, these degree-corrections also suggest other potential frameworks. For example, one motivation to degree-correct a model could be to truly decouple the inferred group identities from the network degrees so that groups are based solely upon the relative bias of their connections rather than their absolute count. Although the generalized model presented in this work mitigates the strong relationship between degree and group found in the non degree-corrected model, it does not fully treat the degree sequence and group structure as *a priori* independent. Future work may address this to formulate a truly unbiased "degree-correction."

2.2 Assortative stochastic block models

In Section 2.1 we discussed how the stochastic block model can be degree-corrected to model degree inequality within groups. In this section we introduce another variant of the usual SBM which directly models the assortative preference for nodes to connect within their own group. This extension enables us to directly measure group assortativity, identify more and smaller groups than the traditional SBM, and improve the predictive performance of the model.

2.2.1 Introduction

As discussed in Section 1.1.2, groups often influence network structure through a *assortative* preference where nodes interact more often within their group than outside of it.

This preference, in fact, is often used to define what a group or a community of nodes in a network means. For example, the modularity measures this tendency by counting the number of edges between groups above what chance predicts. Modularity maximization, one of the most popular community detection methods, therefore defines network groups as the tightly knit clusters of nodes that exhibit the most assortative behavior.

Yet depending on the context of the system, groups can inform network structure in more complex ways than pure assortativity. Some networks may, for example, be disassortative along group lines, as nodes are more likely to interact outside their group than within it. Methods like modularity maximization that search only for assortative structure are unable to identify these types of groups. Furthermore, although nodes of a network may on average exhibit an assortative or disassortative preference, not all groups need to behave identically. Some groups may be assortative while other groups are disassortative. Likewise, certain pairs of groups may be more or less likely to interact with each other than the average pair.

In this work we introduce stochastic block models that capture and measure this rich space of possible group structures in networks. While modularity maximization's narrow definition of network groups precludes it from identifying more complex structures, the expressive models we describe can identify latent groups beyond the scope of more restricted methods.

We build these models upon the usual SBM described in Section 1.2.3. Although this model has some baseline flexibility in its inferences, it is still restricted in a number of ways. For one the model suffers from a *resolution limit* where it cannot identify more than a certain number of groups, roughly $q \sim \sqrt{m}$, even when those groups are clearly distinct. This drawback is shared with modularity maximization, where the resolution limit was first identified [53]. In this work we demonstrate that this limit ultimately stems from an inflexibility of the SBM: that it does not directly incorporate assortativity in its modeling.

The SBM assumes *a priori* that nodes are equally likely to connect both within and outside of their group. Yet, when networks exhibit a strong assortative preference it is possible to identify groups beyond the resolution limit by leveraging knowledge of that assortativity. By generalizing the usual model into an *assortative SBM* we can thus identify these smaller scale groups. Furthermore, by parameterizing this effect the generalized model directly infers how assortative a given network is, providing mechanistic insight into the system it represents.

This type of approach has been considered in other work. For example, the well-studied planted partition (PP) model [48] and recent assortative SBM proposed by Zhang and Peixoto [144] both parametrize this relative preference to connect within versus between groups. In this work, we propose two new assortative models in this lineage. First, a "simple" assortative SBM that generalizes the usual SBM in a more direct manner than previous treatments. Second, a "general" assortative SBM that contains all the other assortative models we consider as special cases.

This general assortative SBM not only parametrizes network assortativity but also how much variation is present in the relationships between groups. Each assortative model we consider makes different assumptions about these variations. For example, the planted partition model assumes all groups are internally identical and that all pairs of groups are equally likely to connect, while the simple assortative model allows for more deviations from this baseline. The general assortative SBM measures how much of this variation occurs within and between groups and therefore which special case best fits a given network.

Applied to a range of real networks, we find strong evidence for our new assortative models and demonstrate that they considerably improve predictive performance over the usual SBM and other assortative models. These new methods allow for a more detailed understanding of networks by being flexible enough in their assumptions to undercover more complex group structures than previously possible.

2.2.2 Resolution limit

The traditional stochastic block model described in Section 1.2.3 is bound by a *resolution limit*. Even if a network "truly" contains many small well-separated communities, the usual SBM is not able to recover them. This resolution limit was first observed [53] as a weakness of the modularity maximization method discussed in Section 1.2.3, but hampers the SBM as well.

This effect can be understood using the microcanonical form Eq. (1.50) of the SBM, reproduced as

$$\begin{aligned} P(\boldsymbol{A}|\boldsymbol{b},\rho) &= \underbrace{\frac{\prod_{r<s} M_{rs}! \prod_r M_{rr}!!/n_r^{m_r}}{\prod_{i<j} A_{ij}! \prod_i A_{ii}!!}}_{\text{multinomial}} \underbrace{\prod_{r<s} \frac{(\rho n_r n_s)^{M_{rs}}}{(\rho n_r n_s + 1)^{M_{rs}+1}} \prod_r \frac{(\rho n_r^2/2)^{M_{rr}/2}}{(\rho n_r^2/2 + 1)^{M_{rr}/2+1}}}_{\text{geometric}} \\ &= P(\boldsymbol{A}|\boldsymbol{M},\boldsymbol{b}) P(\boldsymbol{M}|\boldsymbol{n},\rho). \end{aligned} \tag{2.22}$$

Interpreted as an encoding, the description length of the network thus has two parts

$$H(\boldsymbol{A}|\boldsymbol{b},\rho) = H(\boldsymbol{A}|\boldsymbol{M},\boldsymbol{b}) + H(\boldsymbol{M}|\boldsymbol{n},\rho), \tag{2.23}$$

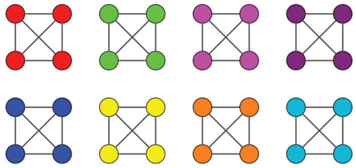
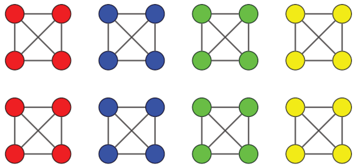
Partition $\boldsymbol{b}$		
Usual SBM $H(\boldsymbol{A}\vert\boldsymbol{b},\rho)$	210.8	**189.8**
Assortative SBM $H(\boldsymbol{A}\vert\boldsymbol{b},\rho_{\text{in}},\rho_{\text{out}})$	**163.0**	172.9

Figure 2.5: Description lengths of an example network given two potential partitions. The "true" partition into the 8 connected components is compared to a coarsening into only 4 groups. The usual SBM prefers the coarsened partition, indicated by the smaller description length, since the resolution limit prevents it from identifying the small groups. The assortative SBM does not have this limitation and finds the true partition.

the transmission of the edge count matrix $\boldsymbol{M}$ and the transmission of the network $\boldsymbol{A}$ given those edge counts. The best-fit group structure found by the SBM is the partition $\boldsymbol{b}$ that minimizes this overall description length, a balancing act between the two terms[3].

To observe the resolution limit, we construct synthetic examples of networks where the "true" partition does not minimize the model's description length, indicating that model fails to recover the underlying group structure. Figure 2.5 contains such an example. In the figure two partitions of the same network of 32 nodes are considered. On the left, the nodes are partitioned into the 8 connected components while on the right these components are paired off into 4 larger groups. The 8 groups are entirely disconnected from each other, and provide clear assortative group structure. Yet, the usual SBM prefers the 4 groups as indicated by the smaller description length and so higher posterior probability of that partition.

This shortfall arises due to how the description length Eq. (2.23) evolves with the number of groups q. Intuitively, the network transmission term requires specifying the placement of each of the m edges at a rough information cost $H(\boldsymbol{A}|\boldsymbol{M},\boldsymbol{b}) \sim O(m)$. To transmit the $q \times q$ edge count matrix, we must specify each of its entries at a cost that scales as $H(\boldsymbol{M}|\boldsymbol{n},\rho) \sim O(q^2)$. Therefore when $q \gtrsim \sqrt{m}$ the cost to transmit the edge count matrix dominates the network term and becomes prohibitively expensive. The model therefore will never report a partition with more than $O(\sqrt{m})$ groups, even when the groups are otherwise apparent. Appendix B.10 contains a more careful treatment of this effect, and places the maximum number of groups at roughly $q \sim \sqrt{m/e}$.

[3] This gives the maximum likelihood partition $\boldsymbol{b}$, the MAP estimate also includes the prior $P(\boldsymbol{b})$, although it does not influence the resolution limit.

In Section 2.2.5 we demonstrate how this resolution limit systematically prevents the stochastic block model from identifying small groups in synthetic tests. Likewise in Section 2.2.6 we show how the usual SBM infers a relatively small number of groups in many real networks.

2.2.3 Simple assortative SBM

In this section we generalize the usual SBM in order to allow the model to overcome the resolution limit. This adjustment is made in a similar spirit to the generalized modularity [115] which overcomes this limit by introducing a parameter to the modularity. In our case, the newly assortative model both avoids this limitation and allows us to measure the assortative preference.

In the usual SBM, the weight matrix entries follow independent and identical exponential distributions with density $\rho > 0$,

$$P(\boldsymbol{\omega}|\rho) = \prod_{r<s} \frac{1}{\rho} e^{-\omega_{rs}/\rho}. \tag{2.24}$$

We define the *simple assortative SBM* by separately distributing the diagonal and off-diagonal weight matrix entries with densities $\rho_{\text{in}}, \rho_{\text{out}} > 0$ respectively as

$$P_{\text{ASBM}}(\boldsymbol{\omega}|\rho_{\text{in}}, \rho_{\text{out}}) = \prod_{r<s} \frac{1}{\rho_{\text{out}}} e^{-\omega_{rs}/\rho_{\text{out}}} \prod_{r} \frac{1}{\rho_{\text{in}}} e^{-\omega_{rr}/\rho_{\text{in}}}. \tag{2.25}$$

These new parameters control the overall density of edges found within and between groups in the network. We thus refer to them as the *in-group density* ρ_{in} and *out-group density* ρ_{out}.

We recover the non-assortative SBM from the assortative model when these densities are equal, $\rho_{\text{in}} = \rho_{\text{out}} = \rho$. To infer the parameters we introduce exponential priors as used for the usual density,

$$P(\rho_{\text{in}}) = e^{-\rho_{\text{in}}}, \quad P(\rho_{\text{out}}) = e^{-\rho_{\text{out}}}. \tag{2.26}$$

Once measured, these parameters give a more detailed picture of network behavior than the overall density ρ alone. For example, the ratio $\rho_{\text{in}}/\rho_{\text{out}}$ represents the overall assortativity, the preference for nodes to connect to others in their group over those outside their group.

Integrating over this new weight matrix distribution, the integrated likelihood of the assortative SBM is

$$\begin{aligned} P(\boldsymbol{A}|\boldsymbol{b}, \rho_{\text{in}}, \rho_{\text{out}}) &= \int P(\boldsymbol{A}|\boldsymbol{\omega}, \boldsymbol{b}) P(\boldsymbol{\omega}|\rho_{\text{in}}, \rho_{\text{out}}) d\boldsymbol{\omega} \qquad (2.27) \\ &= \underbrace{\frac{\prod_{r<s} M_{rs}! \prod_r M_{rr}!!/n_r^{m_r}}{\prod_{i<j} A_{ij}! \prod_i A_{ii}!!}}_{\text{multinomial}} \underbrace{\prod_{r<s} \frac{(\rho_{\text{out}} n_r n_s)^{M_{rs}}}{(\rho_{\text{out}} n_r n_s + 1)^{M_{rs}+1}} \prod_r \frac{(\rho_{\text{in}} n_r^2/2)^{M_{rr}/2}}{(\rho_{\text{in}} n_r^2/2 + 1)^{M_{rr}/2+1}}}_{\text{geometric}} \\ &= P(\boldsymbol{A}|\boldsymbol{M}, \boldsymbol{b}) P(\boldsymbol{M}|\boldsymbol{n}, \rho_{\text{in}}, \rho_{\text{out}}). \qquad (2.28) \end{aligned}$$

This shares the stub-matching network distribution $P(\boldsymbol{A}|\boldsymbol{M}, \boldsymbol{b})$ with the usual SBM although the edge count matrix entries are now geometrically distributed with distinguished means

$$\begin{cases} \mathbf{E} M_{rr} &= \rho_{\text{in}} n_r^2 \\ \mathbf{E} M_{rs} &= \rho_{\text{out}} n_r n_s. \end{cases} \qquad (2.29)$$

The ability for the simple assortative SBM to adapt to highly assortative networks enables it to surpass the resolution limit and identify more groups than the usual SBM. For the disconnected group examples that troubled the usual SBM in Section 2.2.2, we now have the separate in-group density $\rho_{\text{in}} = q\frac{2m}{n^2}$ and out-group density $\rho_{\text{out}} = 0$. For these values, the assortative model assumes that the off-diagonal entries of the edge count matrix vanish, and only models the non-zero diagonal elements M_{rr}.

This significantly reduces the information cost $H(\boldsymbol{M}|\boldsymbol{n}, \rho_{\text{in}}, \rho_{\text{out}})$ to transmit the edge counts in the assortative model. Only the q diagonal elements need to be transmitted at an information cost of $O(q)$, rather than the usual SBM which can not take advantage of this structure and incurs a $O(q^2)$ cost for the entire matrix. The model can therefore report a partition with many more groups without incurring a penalty that dominates the $O(m)$ network transmission. As detailed in Appendix B.10, the assortative SBM is therefore capable of finding many more, smaller groups than the non-assortative model. In Section 2.2.6, we observe that the assortative SBM indeed makes use of this capacity and identifies more groups in real networks than the usual SBM.

2.2.4 General assortative SBM

In the previous section we discussed a simple assortative SBM that allows us to circumvent the resolution limit. In this section we discuss a further extension of the model which also includes alternative assortative SBMs that have been considered elsewhere. By defining this *general assortative SBM* as a nested model, we can directly compare these special cases against each other.

The first of these alternatives is known as the *planted partition* (PP) model. In this widely studied model [48], often itself known as the "stochastic block model," the probability of a connection between nodes depends only on whether those nodes are in same or different groups. That is, the weight matrix is fixed to be

$$\omega_{rs} = \begin{cases} \rho_{\text{in}} & r = s \\ \rho_{\text{out}} & r \neq s. \end{cases} \tag{2.30}$$

This prescription may also be written as a Dirac-delta function prior over weight matrices $\boldsymbol{\omega}$ as

$$P_{\text{PP}}(\boldsymbol{\omega}|\rho_{\text{in}}, \rho_{\text{out}}) = \prod_{r<s} \delta(\omega_{rs} - \rho_{\text{out}}) \prod_{r=1}^{q} \delta(\omega_{rr} - \rho_{\text{in}}). \tag{2.31}$$

This prior leads to different behavior than the SBMs of the last section. Figure 2.6 compares the generative process of each model along with an example of a sampled edge count matrix $\boldsymbol{M}$. In Figure 2.6a, the traditional SBM begins with weight matrix entries $\boldsymbol{\omega}$ independently and exponentially distributed with density ρ that are then Poisson sampled to generate the edge count matrix $\boldsymbol{M}$. This leads to a good deal of variation in the observed edge counts as the weight matrix makes some pairs of groups inherently more or less prone to interact. Comparing the diagonal and off-diagonal elements, this does not result in an overall assortative or disassortative preference.

Figure 2.6b repeats this picture for the simple assortative SBM, where now a higher in-group density ρ_{in} generates an assortative preference in the weight and edge count matrices, along with some variation in those entries. Figure 2.6c compares this to the PP model where the weight matrix $\boldsymbol{\omega}$ is again assortative but the entries are fixed to their expected values. As a result, while the PP model shares the assortative preference of the simple ASBM, the edge count matrix $\boldsymbol{M}$ has far smaller (although non-zero) fluctuations.

To bridge the gap between the PP model, defined by the Dirac-delta prior Eq. (2.31), and the simple ASBM with prior Eq. (2.25),we will use an interpolation strategy similar to the general degree-correction of Section 2.1. We use independent gamma distributions on the weight matrix entries with mean ρ_{in} and *homogeneity* parameter $\lambda_{\text{in}} > 0$ on the

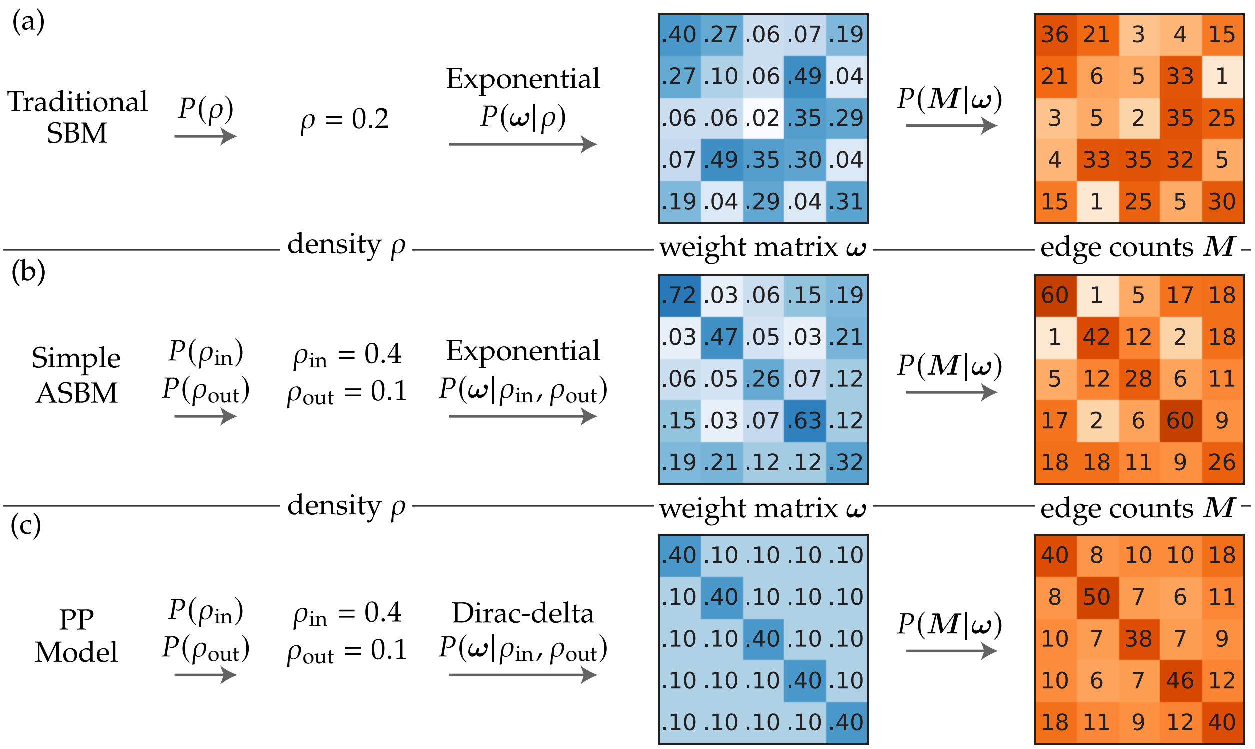


Figure 2.6: Examples of the generative process of the edge count matrix $\boldsymbol{M}$ in the (a) traditional SBM, (b) simple assortative SBM, and (c) planted partition model. In each case the density parameter(s) are first sampled, followed by the weight matrix $\boldsymbol{\omega}$, which generates the observed edge count matrix $\boldsymbol{M}$. Depending on the model, these steps produce final edge counts with and without an assortative preference (high diagonal counts) and to different levels of variation in the entries.

Model	ρ_{in}	ρ_{out}	λ_{in}	λ_{out}	ν_{in}	ν_{out}
Traditional SBM	$\rho_{\text{in}} = \rho_{\text{out}}$		1	1	1/2	1/2
Simple ASBM	inferred	inferred	1	1	1/2	1/2
Hybrid ASBM	inferred	inferred	1	∞	1/2	0
PP model	inferred	inferred	∞	∞	0	0
General ASBM	inferred	inferred	inferred	inferred	inferred	inferred

Table 2.3: Summary of the parameter values of the general assortative SBM presented in this work that correspond to the models it generalizes. In the traditional SBM both the in-group and out-group densities must be the same, $\rho_{\text{in}} = \rho_{\text{out}}$, although this overall density is inferred. For the assortative SBMs and planted partition model, each density is separately inferred from the data. These models are then distinguished by their homogeneity parameters λ_{in} and λ_{out}, which may also be expressed as the variation parameters ν_{in} and ν_{out}.

diagonal, and mean ρ_{out} and parameter $\lambda_{\text{out}} > 0$ off the diagonal[4] as

$$P(\boldsymbol{\omega}|\rho_{\text{in}}, \rho_{\text{out}}, \lambda_{\text{in}}, \lambda_{\text{out}}) = \prod_{r<s} \left(\frac{\lambda_{\text{out}}}{\rho_{\text{out}}}\right)^{\lambda_{\text{out}}} \frac{\omega_{rs}^{\lambda_{\text{out}}-1}}{\Gamma(\lambda_{\text{out}})} e^{-\frac{\lambda_{\text{out}}\omega_{rs}}{\rho_{\text{out}}}} \prod_{r} \left(\frac{\lambda_{\text{in}}}{\rho_{\text{in}}}\right)^{\lambda_{\text{in}}} \frac{\omega_{rr}^{\lambda_{\text{in}}-1}}{\Gamma(\lambda_{\text{in}})} e^{-\frac{\lambda_{\text{in}}\omega_{rr}}{\rho_{\text{in}}}}. \tag{2.32}$$

The parameters λ_{in} and λ_{out} control how homogeneous the weights are on and off of the diagonal. As λ_{in} and λ_{out} diverge, these gamma functions approach Dirac-delta functions and so recover the planted partition model Eq. (2.31) where all weight matrix values are identical on/off the diagonal. If instead $\lambda_{\text{in}} = \lambda_{\text{out}} = 1$ the gamma distributions become exponential distributions, recovering the simple assortative SBM and allowing for some variation in the weight matrix entries. As such we can further recover the traditional SBM by setting the in and out-group densities equal as $\rho_{\text{in}} = \rho_{\text{out}}$. Table 2.3 summarizes the values of these parameters that correspond to the special cases in the general model.

To infer these homogeneity parameters from a network we introduce half-Cauchy priors on the $\lambda_{\text{in}}, \lambda_{\text{out}} > 0$ as

$$P(\lambda_{\text{in}}) = \frac{2}{\pi(\lambda_{\text{in}}^2 + 1)}, \quad P(\lambda_{\text{out}}) = \frac{2}{\pi(\lambda_{\text{out}}^2 + 1)}. \tag{2.33}$$

Once inferred we can also convert these homogeneity measures into the *in-group variation* ν_{in} and *out-group variation* ν_{out} that measure the Gini coefficients of the underlying

[4] In terms of the usual shape and scale parameters of the gamma distribution, the entries are distributed as $\omega_{rr} \sim \Gamma(\lambda_{\text{in}}, \rho_{\text{in}}/\lambda_{\text{in}})$ and $\omega_{rs} \sim \Gamma(\lambda_{\text{out}}, \rho_{\text{out}}/\lambda_{\text{out}})$.

weight matrix entries within and between the groups as

$$v_{\text{in}} = \begin{pmatrix} \lambda_{\text{in}} - 1/2 \\ \lambda_{\text{in}} \end{pmatrix}, \quad v_{\text{out}} = \begin{pmatrix} \lambda_{\text{out}} - 1/2 \\ \lambda_{\text{out}} \end{pmatrix}. \tag{2.34}$$

As shown in Table 2.3, the planted partition model has zero variation $v_{\text{in}} = 0, v_{\text{out}} = 0$ in the weight matrix entries. In contrast the usual and simple assortative SBMs have greater variation $v_{\text{in}} = v_{\text{out}} = 1/2$. By inferring these variation parameters in a network we can assess which of these models fits it best.

Integrating the likelihood Eq. (2.2) over this general weight matrix prior yields

$$P(\boldsymbol{A}|\boldsymbol{b}, \rho_{\text{in}}, \rho_{\text{out}}, \lambda_{\text{in}}, \lambda_{\text{out}}) = P(\boldsymbol{A}|\boldsymbol{M}, \boldsymbol{b})P(\boldsymbol{M}|\boldsymbol{n}, \rho_{\text{in}}, \rho_{\text{out}}, \lambda_{\text{in}}, \lambda_{\text{out}}) \tag{2.35}$$

for the same stub-matching network likelihood $P(\boldsymbol{A}|\boldsymbol{M}, \boldsymbol{b})$ as before and an edge count matrix with negative binomial entries of the same means and parameters,

$$\begin{aligned} P(\boldsymbol{M}|\boldsymbol{n}, \rho_{\text{in}}, \rho_{\text{out}}, \lambda_{\text{in}}, \lambda_{\text{out}}) = \prod_{r<s} &\binom{M_{rs} + \lambda_{\text{out}} - 1}{\lambda_{\text{out}} - 1} \frac{{\lambda_{\text{out}}}^{\lambda_{\text{out}}} (\rho_{\text{out}} n_r n_s)^{M_{rs}}}{(\rho_{\text{out}} n_r n_s + \lambda_{\text{out}})^{M_{rs} + \lambda_{\text{out}}}} \\ \times \prod_{r} &\binom{M_{rr}/2 + \lambda_{\text{in}} - 1}{\lambda_{\text{in}} - 1} \frac{{\lambda_{\text{in}}}^{\lambda_{\text{in}}} (\rho_{\text{in}} n_r^2/2)^{M_{rr}/2}}{(\rho_{\text{in}} n_r^2/2 + \lambda_{\text{in}})^{M_{rr}/2 + \lambda_{\text{in}}}}. \end{aligned} \tag{2.36}$$

Aligning with the special cases, each of this negative binomial distributions reduces to a geometric distribution for parameter $\lambda = 1$ and a Poisson distribution as $\lambda \to \infty$.

This general form of the model also includes another assortative SBM introduced by Zhang and Peixoto [144]. Like the planted partition model and simple ASBM, this model avoids the resolution limit by separately parameterizing the in- and out-group densities. In its non degree-corrected form, the model uses a geometric distribution of edge counts within groups and a Poisson distribution of edge counts between groups. The model is therefore represented by the choice of homogeneity parameters $\lambda_{\text{in}} = 1, \lambda_{\text{out}} = \infty$ or equivalently variations $v_{\text{in}} = 1/2, v_{\text{out}} = 0$.

The model therefore assumes that while there can be variation in the in-group densities, that some groups may be more tightly connected than others, all pairs of distinct groups are equally likely to connect with each other. Since this choice is a mixture of the simple ASBM and the PP model, we will refer to it as the *hybrid ASBM* in our analysis. Since the general ASBM includes the simple ASBM, PP model, and hybrid ASBM as special cases, we can evaluate which of them most appropriately captures the behavior of a given network from the posterior distribution of the variation parameters.

2.2.5 Synthetic tests

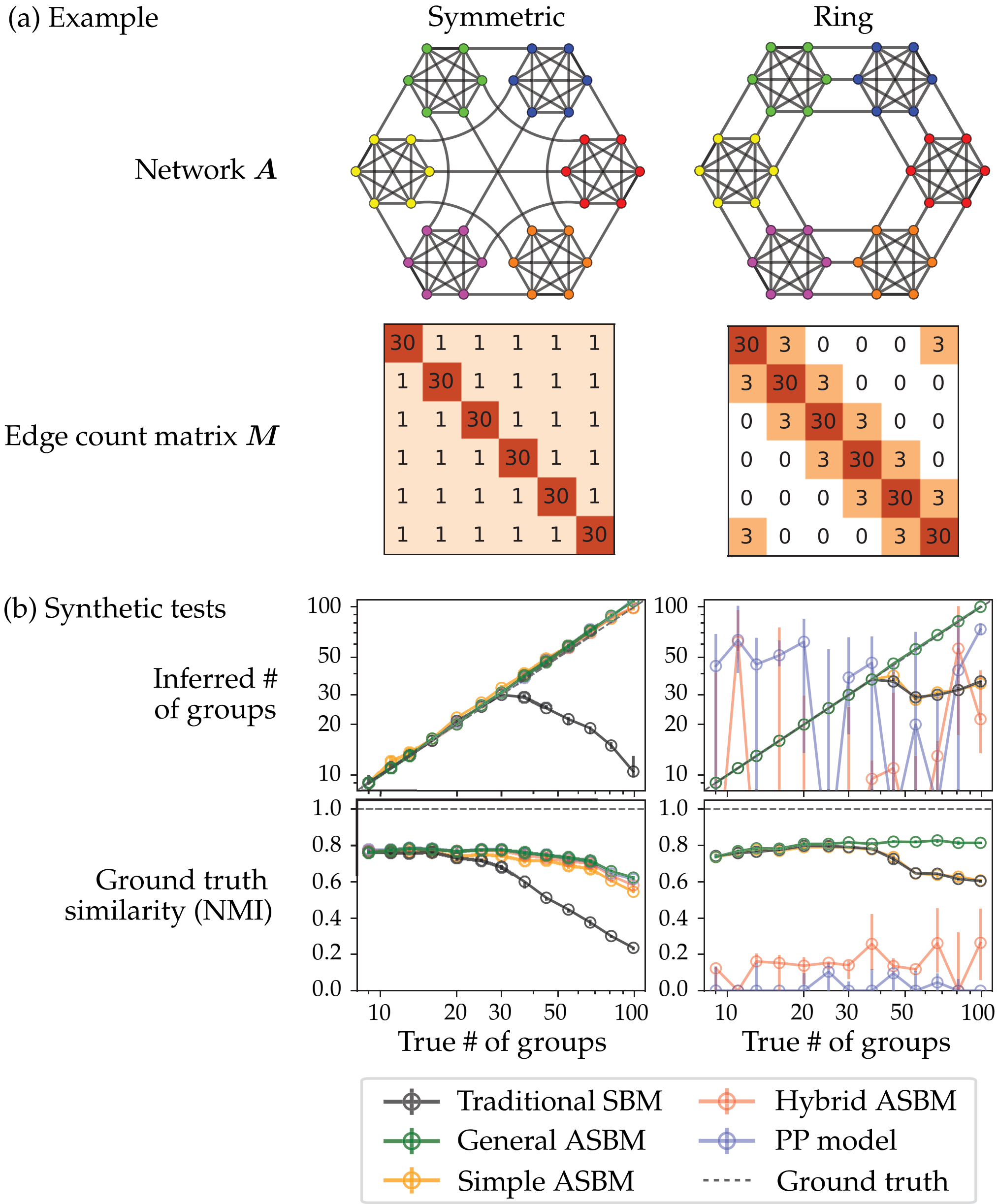
To understand the relative strengths of the models presented in the previous sections we can consider synthetic tests of their performances. In these tests we will construct a network with a "true" planted partition and evaluate how well each model can recover those true groups. We will consider two types of synthetic networks to highlight different aspects of the models.

In the first of these tests, the "Symmetric" case, we partition a network of $n = 1000$ nodes into q equally sized groups. These networks exhibit an assortative preference and so connect more often within than outside the groups. We assign to each group r the same number of internal connections, M_{rr}, and connect each pair of distinct groups r and s with the same number of edges M_{rs}. In terms of our variation parameters, these networks have no variation within or between groups, $v_{\text{in}} = v_{\text{out}} = 0$, like the planted partition model. The total of $m = 2500$ edges are then distributed according to the stub-matching likelihood $P(\boldsymbol{A}|\boldsymbol{M}, \boldsymbol{b})$ to generate a network with average degree $c = 5$ and this symmetric structure $\boldsymbol{M}$.

A smaller example of such a network is given in Figure 2.7a. In our full networks, the ratio of internal to external connections is set such that the signal to noise ratio SNR = 3 is constant, as described in Appendix B.11. This choice is made so that as the number of groups increases, the inherent difficulty of the community detection problem is constant and well above the detectability threshold [6, 42] at SNR = 1 below which recovering the true groups is not possible, even for the planted partition model.

Figure 2.7b demonstrates the results of the various SBMs on these networks as the number of true groups ranges from 10 to 100. For each number of groups we generate 25 synthetic networks and report ranges of performance. To fit each model to a network we use Monte Carlo methods to sample from the posterior distribution of possible group structures and model parameters as detailed in Appendix B.6. In each case the sampled group structures are then summarized using the consensus clustering method of Appendix B.7. To assess model performance, the inferred number of groups is compared against the true count, and the overlap of the found partition with the true group structure is quantified with the normalized mutual information measure described in Chapter 3.

All four assortative models identify a number of groups quite close to the true number across these symmetric tests. The traditional SBM, however, runs into the resolution limit explained in Section 2.2.2 and is unable to identify more than ∼ 30 groups even when many more groups are present. The group truth similarity measures tell a similar story. No model can recover the true groups exactly and achieve a NMI score of 1 due to the inherent uncertainty in their inferences. However, the traditional SBM performs



30	1	1	1	1	1
1	30	1	1	1	1
1	1	30	1	1	1
1	1	1	30	1	1
1	1	1	1	30	1
1	1	1	1	1	30

30	3	0	0	0	3
3	30	3	0	0	0
0	3	30	3	0	0
0	0	3	30	3	0
0	0	0	3	30	3
3	0	0	0	3	30

Figure 2.7: (a) Examples of two types of synthetic networks and their edge count matrices. In the "Symmetric" case all pairs of groups are equally likely to connect while only neighboring groups connect in the "Ring" case. (b) Results of fitting the models presented in this work to synthetic networks of each type as the true number of groups ranges from 10 to 100. Inferred partitions are compared against the truth by comparing the number of groups found and using the normalized mutual information [66]. In both tests the traditional SBM is unable to detect more than ~ 40 groups. In the symmetric case this is caused by the resolution limit while in the ring case it is due to insufficient out-group variation v_{out}. While the planted partition and hybrid assortative models perform well in the symmetric case, they perform very poorly for the ring cases. The general ASBM performs well across all cases considered.

considerably worse than the assortative models as it passes the resolution limit at $q \sim 30$.

To better differentiate between the assortative SBMs that all perform similarly on the symmetric test, we also consider synthetic "Ring" networks with greater variation in their planted group structure. These networks share the same number of nodes and edges as the symmetric examples, and likewise have no variation $v_{\text{in}} = 0$ in the connections within each group. Where these networks differ is that each group only connects to itself and its two neighboring groups in a ring as in Figure 2.7a. This introduces significant variation in the off-diagonal elements of the edge count matrix that increases with the number of groups. The ring networks also differ in having no overall assortativity, $\rho_{\text{in}} = \rho_{\text{out}}$. In this sense the ring networks ask the models to find groups with the opposite strategy as the symmetric test cases. While the symmetric case is structureless apart from the shared assortative in-group preference, the ring case has no such preference, only the non-trivial structure of connections between the groups.

This difference leads the models to very different conclusions in Figure 2.7b. First, the PP model and the hybrid ASBM preform very poorly by either metric since both models assume each pair of distinct groups is equally likely to connect as $v_{\text{out}} = 0$. Since the signal that defines the ring network groups is precisely this out-group variation, neither model can identify meaningful communities. The traditional SBM performs well in many of these cases, as it allows for some amount of variation, assuming $v_{\text{out}} = 1/2$. When the number of communities is large, however, the true variation in the out-group connections is even greater than the traditional model assumes and its performance suffers. We note that this failure is related to, but distinct from, the resolution limit observed in the symmetric tests. The simple ASBM comes to the same conclusions as the traditional SBM since the models are equivalent when $\rho_{\text{in}} = \rho_{\text{out}}$. Only the general ASBM is able to consistently recover the appropriate number of communities as the number of groups and out-group variation increase since the model is flexible enough to infer the appropriate value from the network alone.

2.2.6 Results

In this section we move from synthetic tests to consider the inferences and performances of these assortative models on real network data sets. We apply the models to the same basket of networks considered when degree-correcting the SBM, listed in Table 2.2.

Figure 2.8 plots the number of groups q_{ASBM} found by the simple assortative SBM against the number q_{SBM} identified by the usual SBM along with error bars on these inferences. In some cases the assortative model identifies many more groups than the

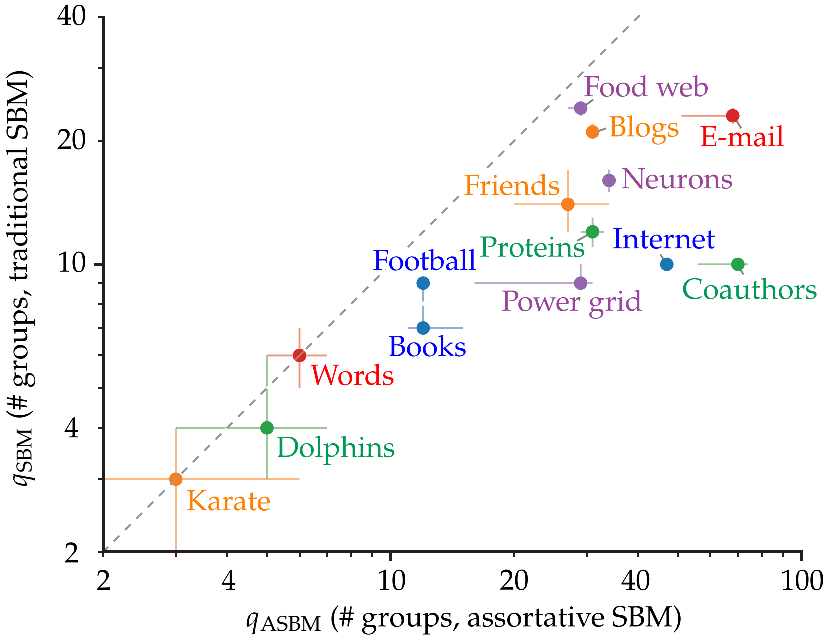


Figure 2.8: Comparison of the number of groups q_{SBM} found by the traditional SBM with the number q_{ASBM} found by the simple assortative SBM in the networks of Table 2.2. The assortative model consistently identifies more groups than the usual SBM, suggesting that the usual SBM is hampered for these networks by the resolution limit. The point colors are meant only to help distinguish the data sets.

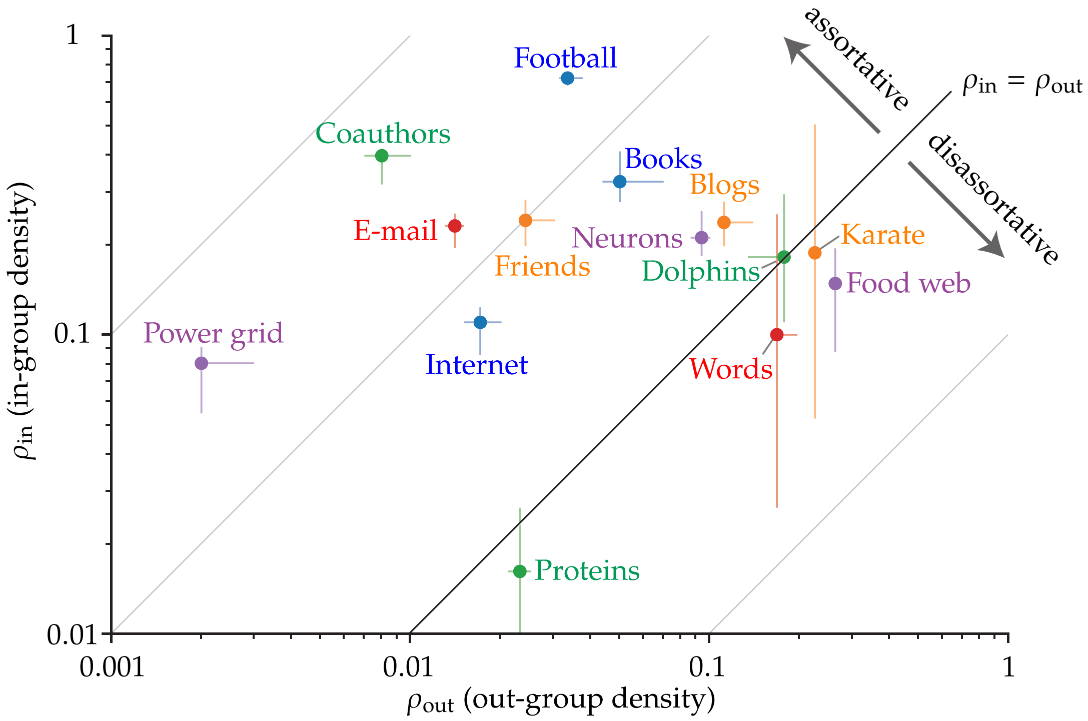


Figure 2.9: Typical in-group and out-group densities inferred by the general assortative SBM for the data sets listed in Table 2.2. Most of the data sets are assortative, $\rho_{\text{in}} > \rho_{\text{out}}$, while some data sets are mildly disassortative, $\rho_{\text{in}} < \rho_{\text{out}}$. Especially in the assortative networks, the posterior distributions of these parameters are well separated from the non-assortative case $\rho_{\text{in}} = \rho_{\text{out}}$ and so exclude the usual non-assortative SBM.

traditional model, particularly when both models agree that many groups are present. This suggests that the usual SBM is prevented from finding the larger true number of groups q_{ASBM} by the resolution limit in these networks, as described in Section 2.2.2.

Figure 2.9 plots the in- and out-group densities ρ_{in} and ρ_{out} found by the simple ASBM. Depending on the network, we observe a wide range of overall assortative preferences $\rho_{\text{in}}/\rho_{\text{out}}$. Some networks like the "Food web," "Proteins," and "Words" data sets exhibit disassortative structure as $\rho_{\text{in}} < \rho_{\text{out}}$. In these applications, the nodes possess group characteristics that lead them to more readily interact outside their group than within it.

In the food web, these groups may map on to trophic levels of the ecosystem where carbon flows are more likely to occur between species of separate levels. In the protein network, proteins may similarly belong to functional groups that typically interact with other proteins of different complementary functions. Likewise the words of our network may belong to grammatical categories like adjectives or nouns that are likely to follow words of a different category. Now, these claims are mere speculations as to how our inferred partitions map onto intuition about these settings. A proper analysis of these individual applications is beyond our expertise and the scope of this work but could be a fruitful application for this type of model.

In the figure we also observe many networks with strongly assortative groups, most notably the "Coauthors," "Power grid," and "Football" examples. Across these cases such assortative behavior could naturally arise from either strong sub-field distinctions, geographic barriers, or conference systems. By allowing the overall assortativity to vary as an inferred parameter, the assortative SBM can directly measure the size of this in-group preference across these settings. In Appendix B.12 we compare this assortativity measure to the modularity of the found partitions, and discuss how the two measurements measure the strength of group structure in contrasting, complementary ways.

The size of this assortative preference also appears to drive much of the disagreement between the numbers of groups found by the traditional and assortative SBMs in Figure 2.8. For example, the usual SBM finds only 10 groups in the highly assortative "Coauthors" network while the assortative model identifies over 70.

Figure 2.9 critically also includes error bars that represent the posterior distributions of the in- and out-group densities. Since the simple ASBM includes the usual SBM as the special case where these densities are equal, $\rho_{\text{in}} = \rho_{\text{out}}$, its posterior distribution demonstrates for which networks the assortative generalization is justified over the non-assortative model. In some networks the posterior density distribution meaningfully intersects with the highlighted non-assortative case, meaning that we cannot reject the usual SBM. For other networks, however, the posterior distribution is well-separated from the line of equality indicating that there is sufficient assortative (or disassortative) signal in the network to justify the more general model.

This leaves us to adjudicate among the assortative models we have considered. As described earlier, this can be done using the posterior distributions of the in- and out-group variation parameters v_{in} and v_{out} within the general ASBM model, since it includes all the assortative models as special cases. Figure 2.10 plots the posterior distributions of the variation parameters found in the same networks of the previous figures. The inferred parameter values differ dramatically depending on the network considered and reveal information about the nature of the underlying group structures.

For example, the "Football" network has little in-group variation at $v_{\text{in}} \approx 0.04$, yet has statistically significant out-group variation $v_{\text{out}} \approx 0.36$. This aligns with our understanding of the collegiate football conference system discussed in Section 1.1.2. Internally each conference operates in roughly the same manner as teams agree to play a fixed number of games, typically 8, within their conference. Out-of-conference games, however, may be

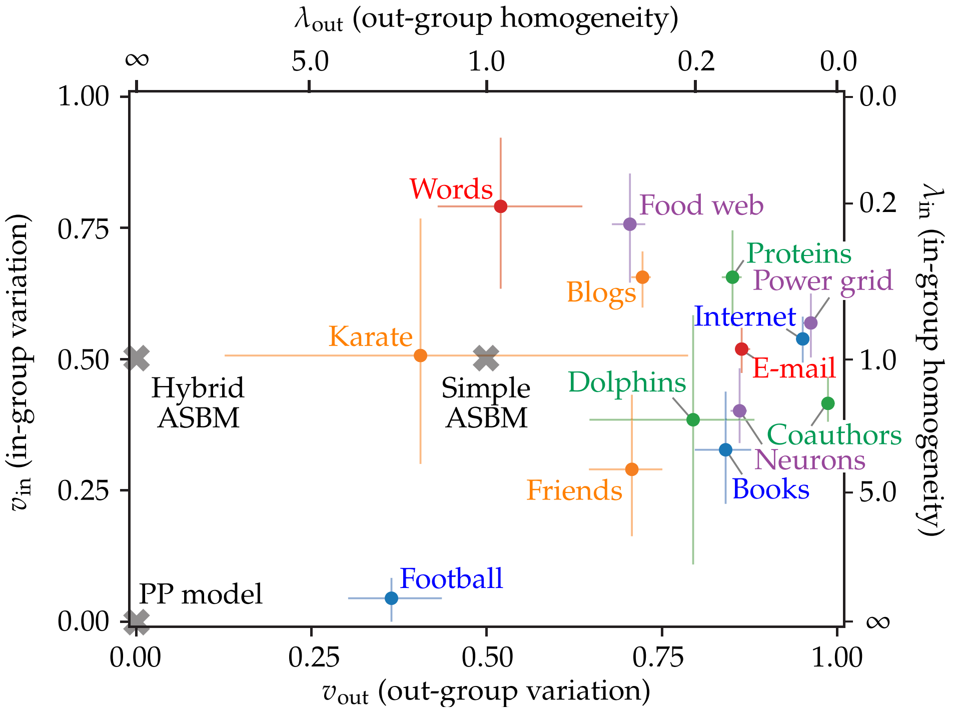


Figure 2.10: Variation in the in-group and out-group weights v_{in} and v_{out} as assessed by the general assortative SBM. The corresponding in and out-group homogeneity parameters λ_{in} and λ_{out} are plotted as well.

more likely to occur between certain pairs of conferences than others.

Although the networks cover a range of variation parameters, their posterior distributions tend to be well-separated from the assortative models, whose parameter values are highlighted by the gray "X"s. Although the simple ASBM is competitive with the full general ASBM for some networks, the hybrid ASBM and PP model are thoroughly excluded for these data sets since they assume no out-group variation, a strong feature of most of these networks.

These general conclusions are also supported by comparing the predictive power of these models in a cross-validation context. In Figure 2.11 the log-posterior-predictive performance of the assortative models over 50 cross-validation splits are plotted relative to the performance of the traditional (non-assortative) SBM. Higher values in this figure indicate that the model can more effectively extrapolate from a randomly chosen 80% subset of the network to the remaining 20%. Across most of these examples we find that the simple ASBM and general ASBM trade positions as the top performing model. As the data sets are arranged from least to most assortative, the figure illustrates that in strongly assortative networks the assortative models naturally gain a significant advantage over the traditional SBM. In most cases we also observe that although the PP model and the

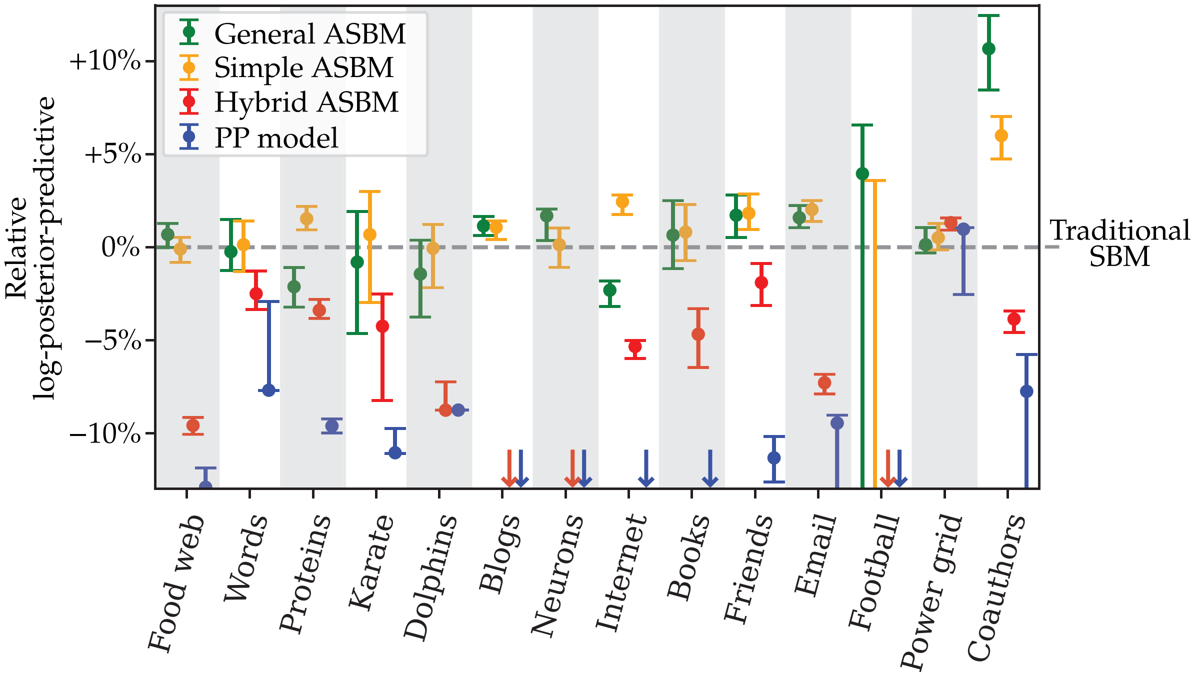


Figure 2.11: Figure of the log-posterior-predictive performance of the assortative models discussed in this work across 50 cross-validation splits of the models relative to the traditional (non-assortative) SBM. Data sets are presented from left to right in order of increasing assortativity $\rho_{\text{in}}/\rho_{\text{out}}$. Downward arrows indicate that model performance is below the plotted range. The general and simple ASBM presented in this work consistently outperform alternative models in these tests, particularly for highly assortative data sets.

hybrid ASBM model this assortativity, their assumptions of no out-group variation hurt their predictive performance, for some data sets considerably.

We also note that the scale of this figure is considerably larger than the corresponding Figure 2.4 describing degree-correction of the SBM. While appropriate degree-correction of the SBM yields improvements of up to 4% in predictive power, the assortative models realize gains over 10% in certain cases. The difference between the best and worst performing assortative models is also much more significant.

2.2.7 Conclusions

In this work we have described a second generalization of the typical stochastic block model that infers both the assortativity of a network and the variation in its group structure. These refinements allow the SBM to find more groups, surpassing the resolution limit, and to perform better on real network data by achieving both more efficient network compression and improved predictive performance.

Since the assortative treatment described in this section is entirely independent of the general degree-correction presented in Section 2.1, both generalizations can be made simultaneously as shown in Appendix B.8. This joint model then inherits the benefits of both model extensions and allows for the simultaneous inference of the many new network parameters. The network examples of this chapter exhibit a wide range of behavior across all these measured dimensions, allowing us to draw sharp contrasts between their structures. We hope that in future work these tools can be applied to better understand the relationship between these networks and broader contexts that inform them.

Chapter 3

Information Theory and Clustering Similarity

In the previous chapter we presented novel methods to identify groups in networks. We evaluated and compared these models in a number of ways, mostly with intrinsic measures like network description length and cross-validation tests. Occasionally we employed a different type of assessment where results are compared to knowledge extrinsic to a network, such as known metadata or "true" planted partitions. For example in Section 2.2.5 we generated synthetic networks alongside known communities, and evaluated models by how well they recover the truth from the network alone. In this context and others, we must quantify the similarity between the found groups and the true groups.

In this chapter, we present an information theoretic measure for comparing generic clusterings. While we apply the framework to the community detection context, it pertains to a much wider class of clustering similarity problems. This measure, known as the mutual information, traditionally has a number of shortcomings. In Section 3.1 we address its bias towards clusterings with too many groups, and in Section 3.2 we resolve a bias introduced in its normalization. These sections are based upon [65] and [66] respectively, both works written in collaboration with Mark Newman and Alec Kirkley. Code implementing our improved measure can be found at https://github.com/maxjerdee/reduced_mutual_information.

3.1 Mutual information of clusterings

Mutual information is commonly used as a measure of similarity between competing labelings of a given set of objects, for example to quantify performance in classification and community detection tasks. As conventionally defined, however, the mutual information can return biased results because it neglects the information cost of the so-called contingency table, a crucial component of the similarity calculation. In principle the bias can be rectified by subtracting the appropriate information cost, leading to the modified measure known as the reduced mutual information, but in practice one can only ever compute an upper bound on this information cost, and the value of the reduced mutual information depends crucially on how good a bound is established. In this section we describe an improved method for encoding contingency tables that gives a substantially better bound in typical use cases, and approaches the ideal value in the common case where the labelings are closely similar, as we demonstrate with extensive numerical results.

3.1.1 Introduction

A common task in data analysis is the comparison of two different labelings of the same set of objects. In this thesis we have focused on community detection, where algorithms attempt to divide a network into groups based on clues in the network structure. We have done this under the presumption that these topological groups are correlated with exogenous properties of network nodes, such as demographics in a social network or chemical function in a biological network. To verify such claims we would like to investigate the similarity between the labels generated by the algorithm and the exogenous labels. But how should one do this, particularly if, as is common, the number of groups can be different between the two labelings?

These comparison questions arise in fields outside of network science as well. How well do demographics predict political affiliation, for example? Or how accurately do blood tests predict clinical outcomes? In each of these cases, we are interested in quantifying the similarity between an experimental labeling and a known "ground truth." Such comparisons are commonly made using the information theoretic measure known as mutual information [34].

Mutual information is a measure of how easy it is to describe one labeling of a set of objects if we already know another labeling. Specifically, it measures how much less information it takes to communicate the first labeling if we know the second versus if we do not. This approach has a number of appealing qualities. It is invariant under permutations of the labels in either labeling, so that labelings do not have to be aligned before comparison. It also generalizes gracefully to the case where the two labelings have different numbers of distinct label values.

On the other hand, the mutual information in its most common form also has some significant drawbacks and, in particular, it is known to be biased towards labelings with a large number of distinct labels. Various proposals have been made for mitigating this issue [44, 131, 145, 9, 101]. In this section we focus on the recently proposed *reduced mutual information* [101], which improves on the standard measure by carefully accounting for subleading terms in the information that are normally neglected.

Any version of the mutual information is an approximation to the true information cost of the labelings being compared. One computes the information cost by defining some encoding scheme for labelings and then counting the number of bits needed to specify a labeling within that encoding. In this section we highlight two common pitfalls that occur when quantifying information cost in this way, which produce errors in opposite directions. The first is the neglect of the cost of certain parts of the transmission process, which thus underestimates the total transmission cost. The standard, unreduced mutual information is an example: it does not include the cost to transmit the "contingency table" that summarizes the relationship between the two labelings, causing it to underestimate—sometimes drastically—the total information cost, particularly for labelings with many groups.

The second pitfall, and the focus of this section, is the use of inefficient encoding schemes, which result in overestimates of information cost. The reduced mutual information, in its conventional form, suffers from this issue because it uses a relatively crude encoding of the contingency table. In this section we offer an improved encoding that gives better bounds on the value of the reduced mutual information, different enough to change outcomes in some practical situations, as we demonstrate with a selection of illustrative examples.

3.1.2 Conditional entropy and mutual information

To motivate our discussion, we first rederive the conventional mutual information using the language of information transmission, before progressing to the reduced mutual information and its variants.

3.1.2.1 Mutual information

Mutual information can be understood in terms of the amount of information required to transmit a labeling from one person to another. As discussed in Appendix A.3, information theory offers a variety of tools to quantify the inherent information content of objects or the information required to transmit the outcomes of probabilistic events.

Suppose, first, that we want to transmit to a receiver a discrete quantity X, which can take any one of a known finite set of N values. For example, we could be transmitting the outcome of a coin flip $X \in \{\text{heads}, \text{tails}\}$ or one possible labeling of a group of objects. If we assign to each possible value of X a unique binary string, we can convey any particular value by transmitting the appropriate string. The minimum length of string that can encode all N values is

$$H = \lceil \log_2 N \rceil \simeq \log_2 N, \tag{3.1}$$

where $\lceil x \rceil$ denotes the smallest integer not less than x. This tells us the number of bits of information needed to transmit X.

Suppose now that X is actually a labeling g of a set of n objects, with each object having exactly one label and each label having the same q_g possible values, which we represent by integers in the range $1 \ldots q_g$. In the community detection context of Chapter 2, this role is played by the vector $\boldsymbol{b}$ that contains the group assignment of each node. The generic labeling g then has $N = q_g^n$ possible values and hence any labeling can be transmitted using an amount of information

$$H(g) = \log N = n \log q_g. \tag{3.2}$$

This, however, is not necessarily the most efficient way to transmit such a labeling. In particular, if different labels occur with different frequencies then a more efficient encoding may be possible, resulting in a smaller information cost. The standard encoding has three parts. First we transmit the number of groups q_g in the labeling. The maximum possible value of q_g is n, so if we use a simple "flat" encoding as in Eq. (3.1), then transmitting any particular value requires information $H(q_g) = \log n$. Next we transmit a vector $n^{(g)}$ of q_g integers $n_r^{(g)}$ equal to the number of objects having each label r. By definition, the $n_r^{(g)}$ sum to n, and the number of ways to choose q_g nonnegative integers that sum to n is $\binom{n+q_g-1}{q_g-1}$, so if we again use a flat encoding to transmit $n^{(g)}$ the information cost will be

$$H(n^{(g)}|q_g) = \log \binom{n + q_g - 1}{q_g - 1}. \tag{3.3}$$

Finally, we transmit the labeling g itself, choosing only from among those that have the correct multiplicities $n_r^{(g)}$ of the labels. The number of such labelings is given by the multinomial $n!/\prod_r n_r^{(g)}!$ and hence, choosing a flat encoding one more time, the information cost is

$$H(g|n^{(g)}) = \log \frac{n!}{\prod_r n_r^{(g)}!}. \tag{3.4}$$

Putting everything together, the information cost, or entropy, to transmit the labeling is

$$\begin{aligned} H(g) &= H(q_g) + H(n^{(g)}|q_g) + H(g|n^{(g)}) \\ &= \log n + \log \binom{n + q_g - 1}{q_g - 1} + \log \frac{n!}{\prod_r n_r^{(g)}!}. \end{aligned} \tag{3.5}$$

This three-step scheme is not the only one that could be applied to this problem, but it is a fairly efficient one in the common case of a small number of groups $q_g \ll n$ with potentially unequal sizes, and it is the one on which the conventional definition of entropy is based. Returning to the network context, we also note that this encoding is equivalent to the prior $P(\boldsymbol{b})$ used over possible group partitions in Eq. (1.47).

The conventional definition, however, ignores all but the last term and approximates the entropy as

$$H_0(g) = \log \frac{n!}{\prod_r n_r^{(g)}!}\,. \tag{3.6}$$

In most cases this is a good approximation. The other terms are subleading contributions—they grow more slowly with n than the leading term—and in practice they are negligible even for quite modest values of n. Commonly one also makes a further approximation, applying Stirling's formula to each of the factorials, which gives the Shannon form of the entropy $H_0(g) = -n \sum_r p_r \log p_r$, where $p_r = n_r^{(g)}/n$ is the probability that a randomly chosen object has label r.

Now consider the corresponding encoding scheme for mutual information. Suppose that we have two different labelings of the same set of objects, a candidate labeling c, generated for instance by some sort of algorithm, and a ground-truth labeling g which represents the "true" labels. The mutual information $I(c;g)$ between the two labelings is defined as the amount of information that can be saved when transmitting the truth g if the receiver already knows the candidate c. We can write this quantity as the total information or entropy needed to transmit g on its own, minus the conditional entropy, the amount to transmit g given prior knowledge of c:

$$I(c;g) = H(g) - H(g|c). \tag{3.7}$$

For the first term, we use the three-part encoding scheme described above, Eq. (3.5). For the second we use a similar multipart scheme, but one that now takes advantage of the receiver's knowledge of c. In this scheme we first communicate q_g and $n^{(g)}$ as before, at the same information cost of $H(q_g) + H(n^{(g)}|q_g)$. Then we communicate a *contingency table* $n^{(gc)}$, a matrix with elements $n_{rs}^{(gc)}$ that represent the number of objects that simultaneously belong to group r in the ground truth g and group s in the candidate labeling c. Figure 3.1 shows an example of a contingency table for two labelings of the nodes of a small network.

The elements of the contingency table are all non-negative integers and its row and

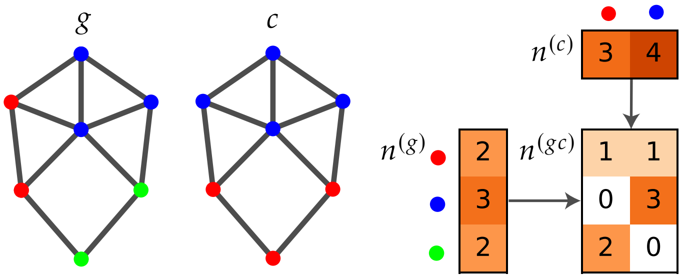


Figure 3.1: A contingency table for two labelings (colors) of the nodes of a network, one with three colors and one with two. The entries in the 3 × 2 contingency table $n^{(gc)}$ count the number of nodes that have each pair of label values. The row and column sums $n^{(g)}$ and $n^{(c)}$ then count the number of nodes with each label in the two labelings. Note that, although we illustrate the contingency table with an application to a network, the table itself is a function of the labelings only and is independent of the network structure.

column sums are equal to the multiplicities of the labels in g and c respectively:

$$\sum_{s=1}^{q_c} n_{rs}^{(gc)} = n_r^{(g)}, \qquad \sum_{r=1}^{q_g} n_{rs}^{(gc)} = n_s^{(c)}. \tag{3.8}$$

Since the receiver already knows the values of $n^{(g)}$ and $n^{(c)}$ (the former because we just transmitted it and the latter because they know c), only contingency tables with these row and column sums need be considered. The information cost to transmit the contingency table with a flat encoding is then equal to $\log \Omega(n^{(g)}, n^{(c)})$, where $\Omega(n^{(g)}, n^{(c)})$ is the number of possible tables with the required row and column sums. There is no known general expression for this number, but approximations exist that are good enough for practical purposes [43, 14, 64].

Finally, after transmitting the contingency table, it remains to transmit the ground-truth labeling itself. For this we need only consider those labelings consistent with the contingency table and the known candidate labeling c. The number of such labelings is $\prod_r n_r^{(c)}!/\prod_{rs} n_{rs}^{(gc)}!$, so the information needed to uniquely identify one of them is

$$H(g|c, n^{(gc)}) = \log \frac{\prod_s n_s^{(c)}!}{\prod_{rs} n_{rs}^{(gc)}!}. \tag{3.9}$$

Putting everything together, the total conditional information is then

$$\begin{aligned} H(g|c) &= H(q_g) + H(n^{(g)}|q_g) + H(n^{(gc)}|n^{(g)}, n^{(c)}) + H(g|c, n^{(gc)}) \\ &= \log n + \log \binom{n + q_g - 1}{q_g - 1} + \log \Omega(n^{(g)}, n^{(c)}) + \log \frac{\prod_s n_s^{(c)}!}{\prod_{rs} n_{rs}^{(gc)}!}. \end{aligned} \tag{3.10}$$

In typical applications the number of labelings compatible with the contingency table is much smaller than the total number of labelings. Thus the amount of information needed to transmit the ground truth using this encoding is typically substantially smaller than Eq. (3.5). The difference—the amount saved—is the quantity we call the mutual information:

$$\begin{aligned} I(c;g) &= H(g) - H(g|c) \\ &= H(q_g) + H(n^{(g)}|q_g) + H(g|n^{(g)}) \\ &\qquad - \left[H(q_g) + H(n^{(g)}|q_g) + H(n^{(gc)}|n^{(g)}, n^{(c)}) + H(g|c, n^{(gc)})\right] \\ &= H(g|n^{(g)}) - H(g|c, n^{(gc)}) - H(n^{(gc)}|n^{(g)}, n^{(c)}) \\ &= \log \frac{n! \prod_{rs} n_{rs}^{(gc)}!}{\prod_r n_r^{(g)}! \prod_s n_s^{(c)}!} - \log \Omega(n^{(g)}, n^{(c)}). \end{aligned} \tag{3.11}$$

Once again this encoding is not necessarily the most efficient one, but it works well in practical situations and forms the basis for the conventional definition of mutual information. And once again the conventional definition drops the subleading term, retaining only the first term in (3.11):

$$I_0(c;g) = \log \frac{n! \prod_{rs} n_{rs}^{(gc)}!}{\prod_r n_r^{(g)}! \prod_s n_s^{(c)}!}. \tag{3.12}$$

Commonly one again also applies Stirling's approximation, which leads to the familiar expression for the mutual information $I_0(c;g) = n \sum_{rs} p_{rs} \log(p_{rs}/p_r p_s)$, where $p_{rs} = n_{rs}^{(gc)}/n$ is the joint probability that a randomly chosen object has label r in the ground truth and s in the candidate labeling.

Although the principles behind them are similar, an important practical difference between Eq. (3.5) for the entropy and Eq. (3.11) for the mutual information is that the subleading term in the mutual information is typically larger and can significantly affect the overall value. It is the neglect of this term that produces the bias towards an excessive number of groups in the conventional mutual information. The cure for this bias is to retain the subleading term, which leads to the measure known as the reduced mutual information.

3.1.2.2 Reduced mutual information

Equation (3.12) defines the standard mutual information I_0, which neglects subleading behavior. In the limit of large n this is a good approximation, but for finite n, including values large enough to be of practical consequence, the subleading term can contribute significantly. In this section, we demonstrate how this gives rise to a bias in favor of labelings with larger numbers of groups and how simply retaining the subleading term removes this bias.

The full expression in Eq. (3.11) is known as the reduced mutual information, with this particular version (we will shortly consider others) distinguished by the fact that it assumes a flat encoding when transmitting the contingency table. We will denote this measure by I_{flat}:

$$I_{\text{flat}}(c;g) = \log \frac{n! \prod_{rs} n_{rs}^{(gc)}!}{\prod_r n_r^{(g)}! \prod_s n_s^{(c)}!} - \log \Omega(n^{(g)}, n^{(c)}). \tag{3.13}$$

The moniker "reduced" derives from the fact that the $-\log \Omega$ term is always negative and so reduces the value of the mutual information relative to the conventional definition of Eq. (3.12), but we emphasize that functionally we are simply retaining terms that are usually neglected. As mentioned previously, there is no general closed-form expression for the number $\Omega(n^{(g)}, n^{(c)})$ of contingency tables with given row and column sums, and its numerical computation is #P-hard and hence intractable for all but the smallest of examples [47]. In practice, therefore, the value must be approximated. In this work we make use of the "effective columns" approximation of [64], which has good performance over a wide range of situations and a simple closed-form expression:

$$\Omega(n^{(g)}, n^{(c)}) \simeq \binom{n + q_c\alpha - 1}{q_c\alpha - 1}^{-1} \prod_{s=1}^{q_c} \binom{n_s^{(c)} + \alpha - 1}{\alpha - 1} \prod_{r=1}^{q_g} \binom{n_r^{(g)} + q_c - 1}{q_c - 1}, \tag{3.14}$$

where

$$\alpha = \frac{n^2 - n + (n^2 - R)/q_c}{R - n}, \quad R = \sum_r (n_r^{(g)})^2. \tag{3.15}$$

This estimate differs from the one originally used for the reduced mutual information in [101], but we favor it here since it performs better in certain regimes.

To appreciate the importance of the contingency table term in the mutual information, consider the simple case where the candidate labeling c places every object in its own group: $c = (1, \ldots, n)$. Regardless of the ground-truth labeling g, this choice of c clearly contains no information about the truth, so we expect the mutual information to be zero. But it is not. We have $n_s^{(c)}! = 1$ for all s in this case, while the contingency table has a single 1 in each column and all other elements are 0, so $n_{rs}^{(gc)}! = 1$ for all r, s, and hence the conventional mutual information of Eq. (3.12) simplifies to

$$I_0(c; g) = \log \frac{n!}{\prod_r n_r^{(g)}!} = H_0(g). \tag{3.16}$$

This answer is as wrong as it possibly could be: we expect the mutual information to take the minimum value of zero, but instead it is equal to the entropy $H_0(g)$, which is its maximum possible value, since the largest amount of information we can save by knowing c when we transmit g is equal to the entire information $H_0(g)$. In other words, the conventional mutual information would have us believe that this candidate labeling which puts every object in its own group tells us everything there is to know about the true labeling g, when in fact it tells us nothing at all.

The reason for this failure is that the contingency table itself uniquely determines g in this case, so neglecting the information content of the table puts the mutual information in error by an amount equal to the complete information cost of the ground truth. If we include the cost of transmitting the contingency table, this erroneous behavior disappears. We can calculate the number $\Omega(n^{(g)}, n^{(c)})$ of contingency tables exactly for this example. Since there is just a single 1 in every column of the table, the number of tables is

$$\Omega(n^{(g)}, n^{(c)}) = \frac{n!}{\prod_r n_r^{(g)}!}, \tag{3.17}$$

and the reduced mutual information is

$$I_{\text{flat}}(c; g) = I_0(c; g) - \log \frac{n!}{\prod_r n_r^{(g)}!} = 0, \tag{3.18}$$

which is now the correct answer.

3.1.2.3 Improved encodings

The reduced mutual information offers a significant improvement over the traditional measure for finite-sized systems, particularly when the candidate labeling has a large number of distinct label values. And, as we have seen, it gives exactly the correct answer in the case where every object is in a group of its own. In this section, however, we argue that the reduced mutual information, as it is usually defined, is itself an imperfect measure, and in particular that it overcorrects for the flaws of traditional mutual information because the encoding scheme used for both $H(g)$ and $H(g|c)$ is inefficient and does not approximate the entropy as well as it could. All calculated entropies are merely upper bounds on the true value: calculating the information cost of transmitting a labeling using a specific encoding guarantees that no more than that amount of information is needed, but it is possible that a better encoding exists that can do the job with less. In this section, we propose more efficient encodings that give better bounds on the entropy and the conditional entropy, particularly in the common case where the two partitions c and g are similar.

The central observation behind our proposed encodings is that quantities like $n^{(g)}$ and $n^{(gc)}$ often have unevenly distributed elements, sometimes strongly so. For example, mutual information is most often used to compare labelings that are quite similar, which means the elements of the contingency table are very non-uniform—those that correspond to common pairs of labels are large, while all the others are small. This in turn means that choices of the contingency table with these properties are much more likely to occur than others and hence that a "flat" encoding that assumes all choices to be equally likely is inefficient. By using an encoding that allows for a non-uniform distribution, we can save a substantial amount of information and achieve a better approximation of the mutual information.

The encodings we propose are based on the Dirichlet-multinomial distribution that had appeared in the microcanonical form of models in Chapter 2. This distribution is central to our current analysis, and so we consider its properties here in further detail. The symmetric Dirichlet-multinomial (DM) distribution is a standard, one-parameter family of discrete distributions over q-vectors X of non-negative integer elements that sum to a given total N. The distribution is derived from a two-part generative process in which, first, a set of q probabilities $p_1 \ldots p_q$ are drawn from a symmetric Dirchlet distribution with concentration parameter $\alpha > 0$, and then a set of q integers $X_1 \ldots X_q$ are drawn from a multinomial distribution with those probability parameters. The resulting distribution over X is given by

$$P(X|N,q,\alpha) = \int \underbrace{\Gamma(q\alpha) \prod_{r=1}^{q} \frac{p_r^{\alpha-1}}{\Gamma(\alpha)}}_{\text{Dirichlet}} \underbrace{N! \prod_{r=1}^{q} \frac{p_r^{X_r}}{X_r!}}_{\text{Multinomial}} \, \mathrm{d}\boldsymbol{p}, \tag{3.19}$$

where the integral is over the simplex of non-negative values p_r such that $\sum_r p_r = 1$. Performing the integral then gives the standard expression for the Dirichlet-multinomial distribution:

$$P(X|N, q, \alpha) = \binom{N + q\alpha - 1}{q\alpha - 1}^{-1} \prod_{r=1}^{q} \binom{X_r + \alpha - 1}{\alpha - 1}. \tag{3.20}$$

If $\alpha = 1$, the Dirichlet-multinomial distribution is uniform over all q-vectors X of non-negative integers that sum to N:

$$P(X|N, q, \alpha = 1) = \binom{N + q - 1}{q - 1}^{-1}. \tag{3.21}$$

Smaller values $0 \leq \alpha < 1$ produce a distribution biased towards more heterogeneous X. In the extreme limit where $\alpha \to 0$ (which we will denote as $\alpha = 0$) the distribution is supported only on vectors that have a single nonzero entry equal to N:

$$P(X|N, q, \alpha = 0) = \begin{cases} 1/q & \text{if } X \text{ has one nonzero entry,} \\ 0 & \text{otherwise.} \end{cases} \tag{3.22}$$

Conversely, for $\alpha > 1$ the Dirichlet-multinomial distribution favors vectors X with more uniform entries, and in the limit $\alpha \to \infty$ it approaches the symmetric multinomial distribution where $p_r = 1/q$ for all r:

$$P(X|N, q, \alpha \to \infty) = \frac{N!}{\prod_{r=1}^{q} X_r!}(1/q)^N. \tag{3.23}$$

Different choices of the parameter α thus place more or less weight on different types of vectors X.

We can use the Dirichlet-multinomial distribution to improve the encoding of the group sizes $n^{(g)}$ and so better approximate the total unconditional information cost $H(g)$. The information cost used in the definition of the standard reduced mutual information is

$$H_{\text{flat}}(g) = H(q_g) + H(n^{(g)}|q_g) + H(g|n^{(g)}), \tag{3.24}$$

where as previously the subscript "flat" indicates the flat encoding that assumes equal probability for all outcomes at each step. In the new approach we propose here, we still use flat encodings for q_g and g but we use a nonuniform Dirichlet-multinomial distribution for the group sizes $n^{(g)}$.

Generally when transmitting a sequence of n values with unequal probabilities such that value r occurs n_r times, the information cost is given by Eq. (3.4):

$$\log \frac{n!}{\prod_r n_r!} \simeq n \log n - n - \sum_r (n_r \log n_r - n_r)$$
$$= - \sum_r n_r \log p_r, \tag{3.25}$$

where $p_r = n_r/n$ is the probability of value r and the factorials are approximated using Stirling's formula. Equation 3.25 tells us that the information cost to transmit the value r is simply $-\log p_r$. Applying this observation to the Dirichlet-multinomial distribution we can calculate the total information cost to transmit a vector $n^{(g)}$ drawn from the Dirichlet-multinomial distribution with concentration parameter α_g:

$$\begin{aligned} H(n^{(g)}|q_g, \alpha_g) &= -\log P(n^{(g)}|q_g, \alpha_g) \\ &= \log \binom{n + q_g\alpha_g - 1}{q_g\alpha_g - 1} - \sum_{r=1}^{q_g} \log \binom{n_r^{(g)} + \alpha_g - 1}{\alpha_g - 1}. \end{aligned} \tag{3.26}$$

The optimal encoding for transmitting $n^{(g)}$ within the Dirichlet-multinomial family is given by the minimum of this expression with respect to α_g, which is also equivalent to simply maximizing $P(n^{(g)}|\alpha_g)$, i.e., to finding the maximum-likelihood value of α_g. In practice we can find the maximum-likelihood value with standard numerical optimization techniques, as described in Appendix C.15.

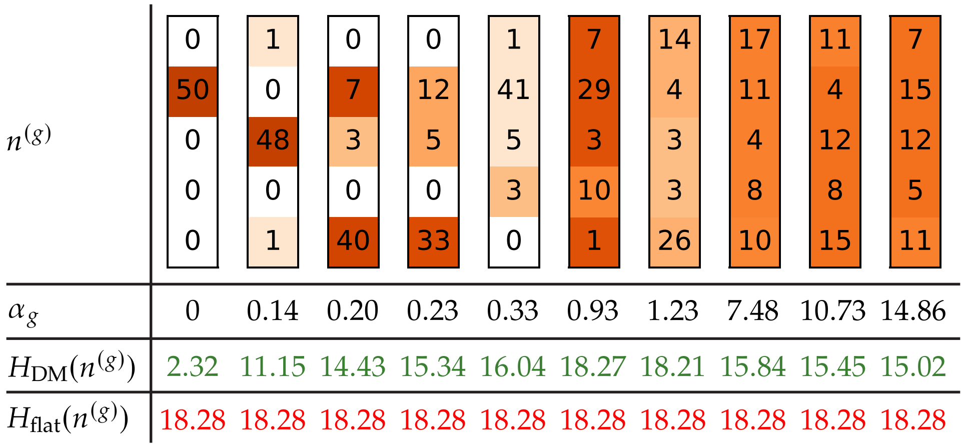
We apply this procedure to a selection of example values of $n^{(g)}$ in Fig. 3.2, giving the optimal values of α_g for each one, along with the resulting values for the entropy. In each case, as we can see, the Dirichlet-multinomial encoding is more efficient than the conventional flat encoding, sometimes by a wide margin. In the extreme case where $n^{(g)}$ has only a single nonzero entry, the optimal value of α_g is zero and the information cost is

$$H(n^{(g)}|q_g, \alpha_g = 0) = \log q_g, \tag{3.27}$$

whereas the cost to transmit the same $n^{(g)}$ using a flat encoding (equivalent to $\alpha_g = 1$) is considerably steeper:

$$H(n^{(g)}|q_g, \alpha_g = 1) = \log \binom{n + q_g - 1}{q_g - 1} \simeq q_g \log n. \tag{3.28}$$

One could argue that to truly make a fair comparison, one should also include the cost



$n^{(g)}$	0	1	0	0	1	7	14	17	11	7
	50	0	7	12	41	29	4	11	4	15
	0	48	3	5	5	3	3	4	12	12
	0	0	0	0	3	10	3	8	8	5
	0	1	40	33	0	1	26	10	15	11
α_g	0	0.14	0.20	0.23	0.33	0.93	1.23	7.48	10.73	14.86
$H_{\text{DM}}(n^{(g)})$	2.32	11.15	14.43	15.34	16.04	18.27	18.21	15.84	15.45	15.02
$H_{\text{flat}}(n^{(g)})$	18.28	18.28	18.28	18.28	18.28	18.28	18.28	18.28	18.28	18.28

Figure 3.2: Optimal values of α_g for the transmission of the community sizes $n^{(g)}$, along with the resulting information cost in bits $H_{\text{DM}}(n^{(g)})$ in our new Dirichlet-multinomial encoding scheme and the corresponding cost $H_{\text{flat}}(n^{(g)})$ in the old, flat encoding. Note how vectors with more extreme values benefit from smaller values of α_g, while more uniform vectors favor larger α_g.

to transmit the value of α_g itself. As shown in Appendix C.15, however, this cost is small in practice, and moreover cancels completely from the final value of the mutual information, so it is normally safe to ignore it, as we do here.

Although the information saved by using the Dirichlet-multinomial distribution is in some cases a significant fraction of the information needed to transmit $n^{(g)}$, it is normally quite small next to the information needed to transmit the entire labeling, which is dominated by the cost $H(g|n^{(g)})$ of sending the labeling itself. The same is not true, however, when we turn to the conditional entropy $H(g|c)$, where using the Dirichlet-multinomial distribution can result in large efficiency gains, and this is our primary motivation for taking this approach.

Recall that the standard model for the conditional information breaks the transmission process into four steps—transmission of q_g, $n^{(g)}$, $n^{(gc)}$, and g—which can be represented by the equation

$$H_{\text{flat}}(g|c) = H(q_g) + H(n^{(g)}|q_g) + H(n^{(gc)}|n^{(g)}, n^{(c)}) + H(g|c, n^{(gc)}), \tag{3.29}$$

with a flat encoding at each step. In our alternate proposal, we again transmit q_g using a flat encoding, but then combine the second and third steps to transmit the contingency table all at once, given q_g and $n^{(c)}$. This transmission again leverages a nonuniform encoding to achieve efficiency gains. The final step of transmitting g itself remains unchanged.

Our process for transmitting the contingency table involves transmitting one column at a time using the Dirichlet-multinomial distribution. We use the same value $\alpha_{g|c}$ for each column, but the columns are otherwise independent. If we denote column s by $n^{(gc)}_{\cdot s}$, then the information cost of this procedure can be written

$$\begin{aligned} H(n^{(gc)}|n^{(c)}, q_g, \alpha_{g|c}) &= \sum_{s=1}^{q_c} H(n^{(gc)}_{\cdot s}|n^{(c)}_s, q_g, \alpha_{g|c}) \\ &= \sum_{s=1}^{q_c} \left[\log \binom{n^{(c)}_s + q_g\alpha_{g|c} - 1}{q_g\alpha_{g|c} - 1} - \sum_{r=1}^{q_g} \log \binom{n^{(gc)}_{rs} + \alpha_{g|c} - 1}{\alpha_{g|c} - 1} \right]. \end{aligned} \tag{3.30}$$

For instance, consider the special case of perfect recovery where $c = g$, so that $n^{(gc)}$ is a diagonal matrix and each column $n^{(gc)}_{\cdot s}$ has only a single nonzero entry. Then, as in Eq. (3.27), $\alpha_{g|c} = 0$ is the optimal choice for transmitting the contingency table and the total information cost is simply

$$H(n^{(gg)}|n^{(g)}, \alpha_{g|c} = 0) = \sum_{r=1}^{q_g} H(n^{(gg)}_{\cdot r}|n^{(g)}_r, \alpha_{g|c} = 0) = \sum_{r=1}^{q_g} \log q_g = q_g \log q_g. \tag{3.31}$$

In the traditional flat encoding the cost is much greater:

$$\begin{aligned} H(n^{(gg)}|n^{(g)}) = H(n^{(g)}|n, q_g) + H(n^{(gc)}|n^{(c)}, n^{(g)}) = \log \binom{n-1}{q_g - 1} + \log \Omega(n^{(c)}, n^{(q)}) \\ = \mathrm{O}(q_g q_c \log n). \end{aligned} \tag{3.32}$$

This significant improvement also extends to the case where $c \simeq g$ and the labelings are similar but not identical. It is in precisely this regime that mutual information is often applied to quantify similarity, so the new encoding is much preferred over the old one in common settings, a conclusion strongly confirmed by the example applications given in Section 3.1.3.

Figure 3.3 shows the equivalent of Fig. 3.2 for the transmission of a selection of example contingency tables. In the case where the candidate labeling places all objects in a single group, so that $q_c = 1$, our proposed scheme is exactly analogous to our method for transmitting the vector $n^{(g)}$, which implies that the mutual information is $I_{\mathrm{DM}}(c;g) = 0$ (the DM denoting "Dirichlet-multinomial"). This is a desirable property which is also shared by the traditional and reduced mutual informations—a candidate labeling that places all objects in a single group tells us nothing about the ground truth g. We also note the considerable gulf in efficiency between the two encodings for the case of identical labelings $g = c$, the second column in Fig. 3.3, while for labelings that are dissimilar (the

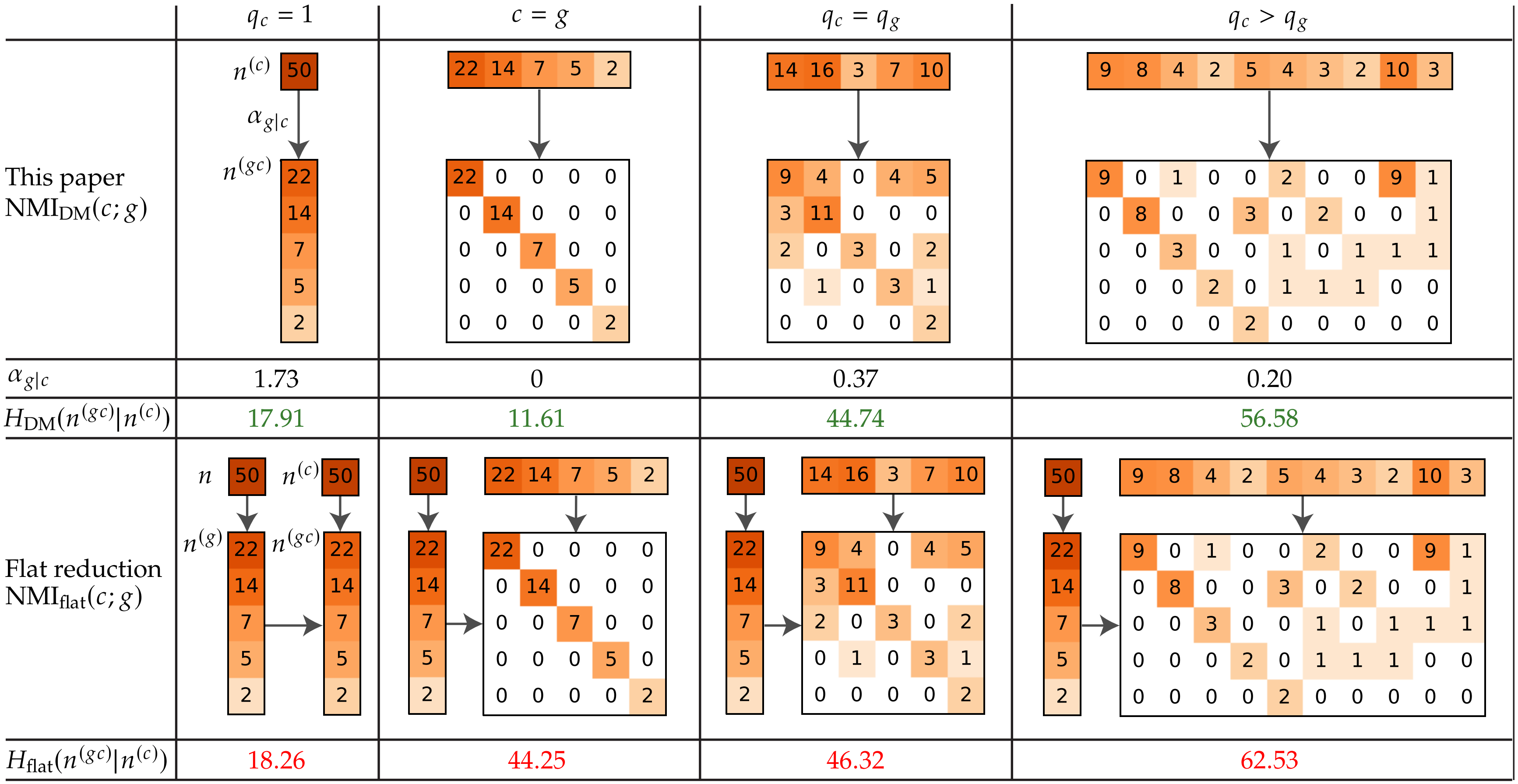


Figure 3.3: Comparison of the information cost of transmitting example contingency tables under the old and new encodings. The top half of the figure shows the new encoding, along with the values for the optimal Dirichlet-multinomial parameter $\alpha_{g|c}$ and the resulting information cost $H_{\mathrm{DM}}(n^{(gc)}|n^{(c)})$. The bottom half shows the old encoding and associated information cost $H_{\mathrm{flat}}(n^{(gc)}|n^{(c)})$. The new encoding is more efficient in every case, and especially so in the case of equal labelings $c = g$.

final two columns of the figure), the gains of the new encoding are more modest, as we would expect.

Employing our new encoding in the calculation of the conditional information, we now obtain a revised information cost of

$$H_{\mathrm{DM}}(g|c) = H(q_g) + H(n^{(gc)}|n^{(c)}, q_g, \alpha_{g|c}) + H(g|c, n^{(gc)}). \tag{3.33}$$

(Once again one could arguably also include the fixed cost of transmitting the value of $\alpha_{g|c}$, but in practice this cost is small and moreover cancels from the final value of the mutual information—see Appendix C.15.)

Putting everything together, we then arrive at our improved mutual information measure

$$\begin{aligned} I_{\mathrm{DM}}(c;g) &= H_{\mathrm{DM}}(g) - H_{\mathrm{DM}}(g|c) \\ &= I_0(c;g) + H(n^{(g)}|q_g, \alpha_g) - H(n^{(gc)}|n^{(c)}, q_g, \alpha_{g|c}) \\ &= I_0(c;g) + \log \binom{n + q_g\alpha_g - 1}{q_g\alpha_g - 1} - \sum_{r=1}^{q_g} \log \binom{n_r^{(g)} + \alpha_g - 1}{\alpha_g - 1} \\ &\qquad - \sum_{s=1}^{q_c} \log \binom{n_s^{(c)} + q_g\alpha_{g|c} - 1}{q_g\alpha_{g|c} - 1} + \sum_{r=1}^{q_g}\sum_{s=1}^{q_c} \log \binom{n_{rs}^{(gc)} + \alpha_{g|c} - 1}{\alpha_{g|c} + 1}. \end{aligned} \tag{3.34}$$

Besides giving an improved estimate of the mutual information, this formulation has a number of additional advantages over the standard reduced mutual information. The closed-form expression means that approximations like those used for the number $\Omega(n^{(g)}, n^{(c)})$ of contingency tables in Eq. (3.14) are unnecessary—the measure can be calculated exactly without approximation. Another advantage is that it is possible to prove that $I_{\mathrm{DM}}(c;g) \leq I_{\mathrm{DM}}(g;g)$ for all c, with the exact equality holding only when c and g are identical up to a permutation of labels, an intuitive result that is required for proper normalization, but which has not been shown for the standard reduced mutual information. We give the proof in Appendix C.13.

This being said, the encoding we use is not necessarily the last word in calculation of the mutual information. As discussed at the start of this section, all entropy calculations only give bounds on the true value and it is possible that another encoding exists that gives a better bound. One could imagine trying an approach analogous to that used for the standard reduced mutual information and constraining not only the column sums of the contingency table but also the row sums, while still using a nonuniform distribution over tables subject to these constraints. Placing more constraints on the contingency table should reduce the number of tables we need to consider and hence save on transmission costs. This approach, however, turns out to offer little benefit in practical situations because the gains must be offset against the information needed to transmit the row sums. It transpires that, in the common case where the candidate and ground-truth labelings are similar to one another, the possible values of the row sums are already tightly restricted, even without placing any explicit constraint on them, so that imposing such a constraint saves little information, while the cost of transmitting the row sums is considerable. In most cases, therefore, this approach is less efficient than the one we propose.

Equation 3.34 does still have some shortcomings. For one thing, it is not fully analytic, since the values of the parameters α_g and $\alpha_{g|c}$ must be found by numerical optimization (see Appendix C.15). Also, because of the asymmetric encoding used to capture the contingency table, in which rows and columns are treated differently, the measure is not symmetric under interchange of c and g. For typical applications where one is comparing candidate labelings against a single ground truth this does not matter greatly, since the problem is already inherently asymmetric, but there may be cases where a symmetric measure would be preferred. Lastly, the encoding we propose is not guaranteed to always perform better than the standard (flat) reduced information. In particular, if the labelings c and g are truly drawn from the distribution corresponding to the flat encoding scheme, then by definition the flat mutual information will give an optimal encoding and our method cannot do better. Our broader claim, however, is that in the realistic regime of labelings that have a significant degree of similarity, our new encoding can be expected to perform better than the flat encoding.

3.1.2.4 Normalized mutual information

So far we have defined various measures of absolute information content, as quantified in bits for example, but such absolute measures can be difficult to interpret. Are 20 bits of mutual information a little or a lot? To make sense of the results, they are often expressed in terms of a normalized mutual information (NMI) that represents the information content as a fraction of its maximum possible value [36]. There are various ways to perform the normalization [88]. Here we use the form

$$\text{NMI}(c;g) = \frac{I(c;g)}{I(g;g)}. \tag{3.35}$$

Note that this expression is asymmetric in c and g—the value is not invariant under their interchange. Other normalized mutual information measures use a symmetric denominator, such as $\frac{1}{2}[I(g;g) + I(c;c)]$ [36, 88], giving a normalized measure that is itself symmetric.

In Section 3.2 and [66] we discuss how these symmetric normalizations introduce bias to the mutual information, and so we will prefer the asymmetric normalization in this analysis.

We can define a normalized mutual information of the form (3.35) for any of the mutual information measures discussed in this chapter, including the Dirichlet-multinomial measure. All of the resulting versions of NMI have the desirable properties of being 1 when $c = g$ and zero when the mutual information is zero. Thus NMI values approaching 1 generally indicate similar labelings and values near zero indicate dissimilar ones, making this an intuitive measure of similarity. It is also possible for the NMI to become (slightly) negative for the reduced mutual information measures we consider [101]. (This cannot happen with the traditional unreduced mutual information.) A negative value indicates that the encoding scheme that makes use of c when transmitting g is actually less efficient than simply transmitting g alone. This, however, happens only when c and g are very dissimilar and hence rarely occurs in practical situations (where we are usually concerned with candidates c that are similar to the ground truth).

For the particular case of the Dirichlet-multinomial mutual information, the NMI has the additional desirable property that it takes the value 1 if and only if c and g are identical up to a permutation of labels, while for all other c it is less than 1—see Appendix C.13. This follows directly from the inequality $I_{\mathrm{DM}}(c;g) \leq I_{\mathrm{DM}}(g;g)$ mentioned above. The same is not true of the conventional unreduced NMI, defined as in Eq. (3.35), which can be 1 even for very dissimilar labelings (see Section 3.1.2.2). It is potentially true, but currently unproven, for the flat reduced mutual information.

3.1.3 Example applications

In this section we give a selection of example applications of our proposed measure, demonstrating that it can give significantly different answers from previous measures—different enough to affect scientific conclusions under real-world conditions.

3.1.3.1 Comparison of the proposed measure and the standard reduced mutual information

In some circumstances the results returned by the measure proposed in this section can diverge significantly from those given by either the non-reduced mutual information or the standard ("flat") version of the reduced mutual information. We have already seen examples for the non-reduced measure: cases in which the candidate labeling c has many

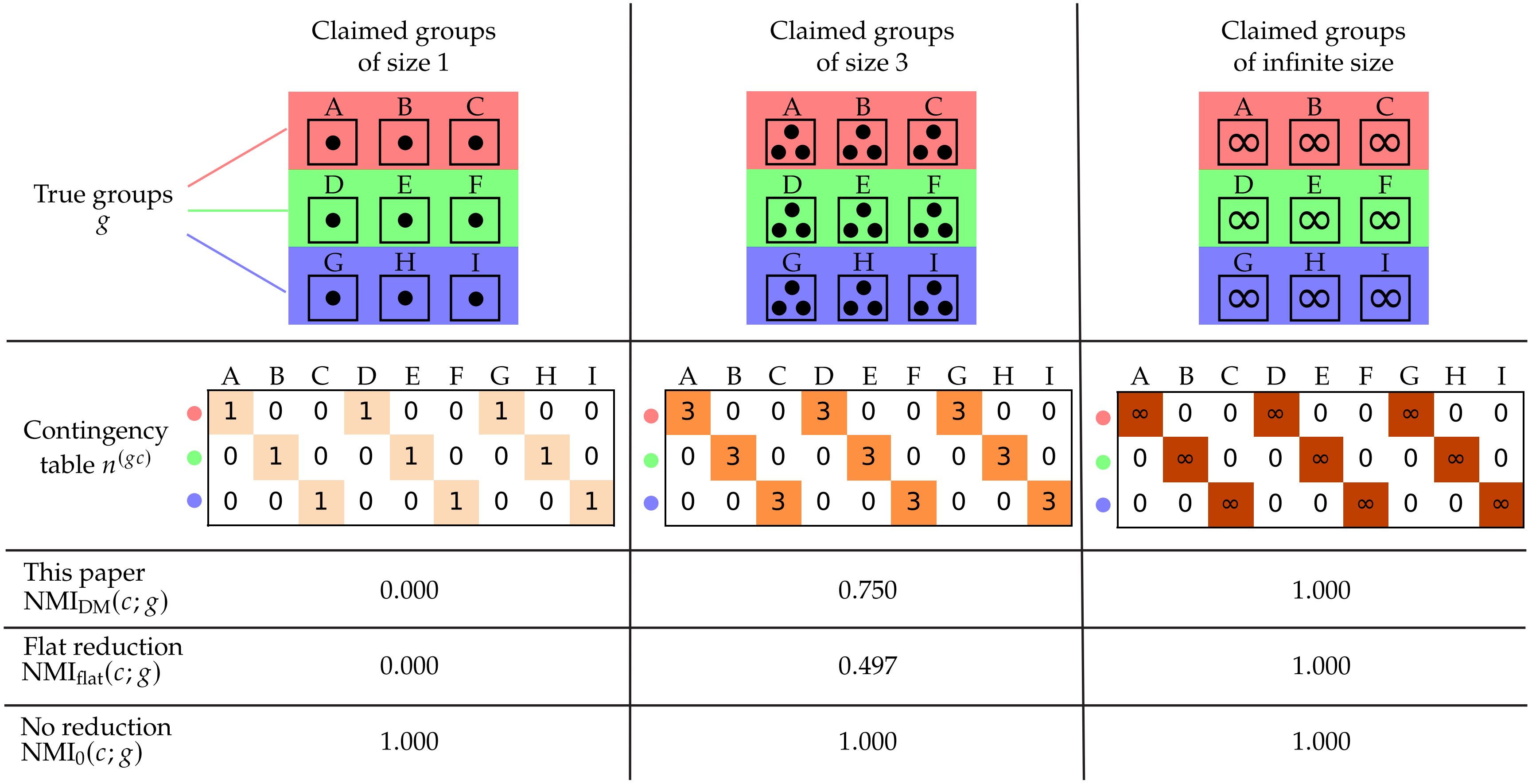


Figure 3.4: An example where the three normalized mutual information measures considered here differ substantially. A set of objects (circular dots) is divided into three ground-truth groups, represented by the horizontal stripes of red, green, and blue. Three competing candidate divisions of the same objects are represented by the black boxes denoted A to I. In each, the ground-truth groups are divided into a set of equally sized subgroups. But regardless of how many objects are in each subgroup, the unreduced measure NMI_0 returns a maximal score of 1 for all the candidate divisions, while the reduced measures rightfully return lower scores, except in the case where the sizes of the subgroups diverge. For subgroups of intermediate size, such as the groups of size three in the middle column, the Dirichlet-multinomial measure $\mathrm{NMI}_{\mathrm{DM}}$ of this section can give a substantially different and larger score than the standard ("flat") reduced mutual information $\mathrm{NMI}_{\mathrm{flat}}$.

more labels than the ground truth can cause the unreduced measure to badly underestimate the true information cost, sometimes maximally so—see Section 3.1.2.2.

A simple example illustrating the difference between the Dirichlet-multinomial and flat versions of the reduced mutual information is shown in Fig. 3.4. In this example a set of objects, represented by the dots in the figure, are split into three equally sized ground-truth groups. The candidate labeling c respects this division but further splits each of the three groups into three subgroups, also of equal size. This is a special case of the situation mentioned above in which the candidate division has more labels than the ground truth, so it comes as no surprise that the conventional, unreduced mutual information overestimates similarity in this case—indeed it returns the maximal value of 1.

To understand the behavior of our two reduced mutual information measures we consider three special cases. In the first, shown in the left column of Fig. 3.4, each of the subgroups, labeled A to I, has size 1, meaning that every group in c has only a single object in it. We discussed this case previously in Section 3.1.2.2 and argued that the correct mutual information should be zero. As the figure shows, both versions of the reduced mutual information give this correct result, while the unreduced measure is maximally incorrect.

Next, consider the right column of Fig. 3.4, which shows what happens as the total number of objects tends to infinity and the size of the subgroups A to I diverges. Asymptotically, the candidate c now gives full information about the ground truth—g is fully specified when both c and the contingency table are known, but the information cost of transmitting the contingency table is a vanishing fraction of the total. Thus the NMI should be 1 in this case, and again both versions of the reduced mutual information give the right answer. (In this limit the unreduced measure also gives the right answer.)

But now consider the middle column of the figure, in which subgroups A to I have size 3. In this case the contingency table, as shown in the figure, is highly non-uniform, and hence is transmitted much more efficiently by the Dirichlet-multinomial encoding than by the flat encoding. This produces a substantial difference between the values of the two reduced mutual information measures. The Dirichlet-multinomial measure gives a relatively high value of 0.75, indicating a strong similarity between candidate and ground truth, while the standard flat measure gives a significantly smaller value, less than 0.5. This is a case where the standard measure has penalized the mutual information too heavily by overestimating the information content of the contingency table, thereby giving a misleading impression that the two labelings are more dissimilar than in fact they are.

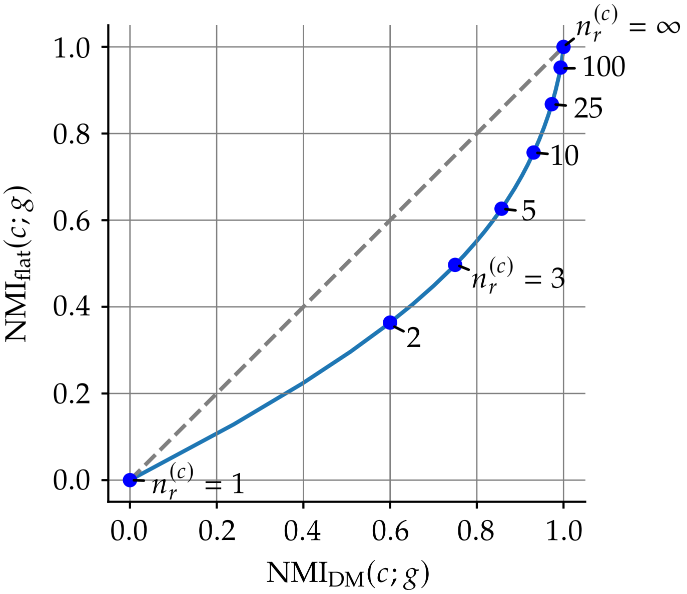
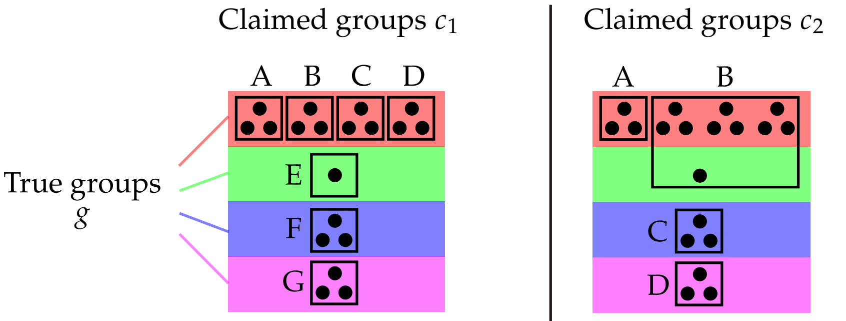


Figure 3.5: Comparison of the two normalized reduced mutual information measures considered here for the example system in Fig. 3.4, for various sizes $n_r^{(c)}$ of the subgroups as denoted by the labels. In all cases except $n_r^{(c)} = 1$ and ∞ the flat reduced mutual information $\mathrm{NMI}_{\mathrm{flat}}$ returns a lower value than the Dirichlet-multinomial reduced mutual information $\mathrm{NMI}_{\mathrm{DM}}$ because it overestimates the information cost of the contingency table.

Figure 3.5 shows a plot of the difference between the two reduced measures for this example system with subgroup sizes $n_r^{(c)}$ ranging all the way from 1 to ∞. Across the entire range we observe that, apart from the limiting values of $n_r^{(c)} = 1$ and ∞, the flat reduced mutual information consistently gives underestimates relative to the Dirichlet-multinomial measure.

Figure 3.6 shows a different aspect of the two reduced measures. In this example the ground-truth labeling divides a set of 19 objects into four groups of varying sizes, and we compare outcomes for two proposed candidate labelings, denoted c_1 and c_2. Labeling c_1 has identified the four groups correctly but splits one of them into a further four subgroups. As we would expect, the conventional unreduced NMI awards this labeling a maximal score of 1, which is clearly incorrect. Both reduced measures correctly give a value less than 1, although the values are somewhat different.

Now consider candidate labeling c_2, which erroneously amalgamates the second ground-truth group with part of the first as shown. Most observers would probably say that this labeling is worse than c_1, but that is not what the standard reduced mutual information reports: the standard measure favors c_2 over c_1 by a substantial margin. On



Contingency table $n^{(gc)}$

	A	B	C	D	E	F	G
🟥	3	3	3	3	0	0	0
🟩	0	0	0	0	1	0	0
🟦	0	0	0	0	0	3	0
🟪	0	0	0	0	0	0	3

	A	B	C	D
🟥	3	9	0	0
🟩	0	1	0	0
🟦	0	0	3	0
🟪	0	0	0	3

This paper $\mathrm{NMI}_{\mathrm{DM}}(c; g)$	0.762	0.646
Flat reduction $\mathrm{NMI}_{\mathrm{flat}}(c; g)$	0.548	0.673
No reduction $\mathrm{NMI}_0(c; g)$	1.000	0.854

Figure 3.6: Two candidate labelings of the same set of objects. In this figure, 19 objects are divided among four ground-truth groups, represented by the horizontal stripes of red, green, blue, and magenta, and two candidate labelings c_1 and c_2 are denoted by the boxes labeled A to G. The Dirichlet-multinomial measure of this section favors the left labeling c_1 while the flat reduced mutual information prefers the right one c_2.

the other hand, the Dirichlet-multinomial measure of this section correctly favors c_1, by a similar margin.

3.1.3.2 Network community detection

The examples of the previous section are illustrative but anecdotal. To shed more systematic light on the performance of the new measure we apply it to the outcomes of a large set of network community detection calculations. In these tests we use the popular Lancichinetti-Fortunato-Radicchi (LFR) graph model [77] to generate 100 000 random networks with known community structure and realistic distributions of node degrees and group sizes. Then we use six different popular community detection algorithms to generate candidate divisions of these networks, which we compare to the known structure

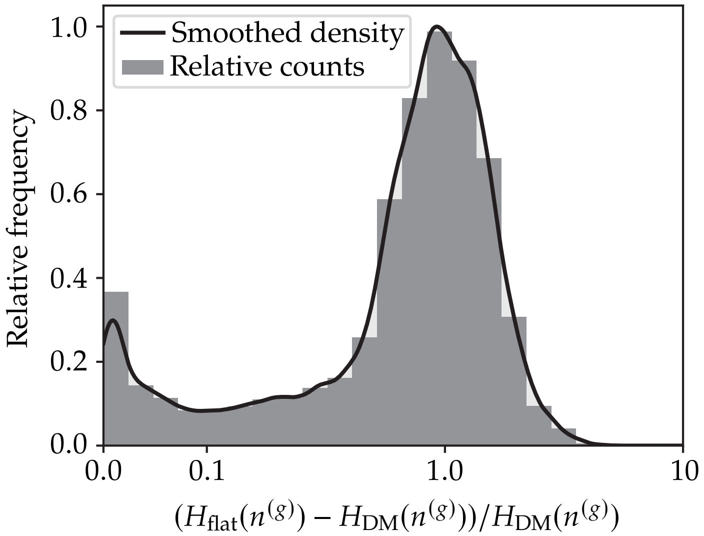


Figure 3.7: Fractional reduction in information cost when transmitting the group sizes $n^{(g)}$ under the Dirichlet-multinomial encoding versus the flat encoding. The horizontal axis is linear from 0 to 0.01 and logarithmic above 0.01. The bar chart is a simple histogram of the relative frequencies, while the curve shows the same results smoothed using a quartic kernel density estimator with the same bin width.

using both the conventional reduced mutual information and the measure proposed here. Further details on the LFR network generation process are given in Appendix C.16.4, and the community detection methods are described in Appendix C.16.2.

As discussed in Section 3.1.2.3, our Dirichlet-multinomial approach improves the efficiency of information transmission in two places: in the transmission of the group sizes $n^{(g)}$ and the transmission of the contingency table $n^{(gc)}$. Figure 3.7 shows the fractional improvement in information cost for the group sizes for each of our test networks. The gains vary substantially between networks, and some are close to zero, but in a large fraction of cases they reach 10% or more.

More important, however, are the gains in transmission of the contingency table. Since the contingency table depends on the candidate c as well as the ground truth, these gains also depend on the candidate and hence vary between the six different community detection algorithms, but for our purposes here we aggregate the results over algorithms. Figure 3.8a shows the distribution of the resulting fractional information savings for all networks in a single plot. The different curves in the plot show how the distribution varies as a function of how similar the ground-truth and candidate divisions are, measured using the Dirichlet-multinomial NMI.

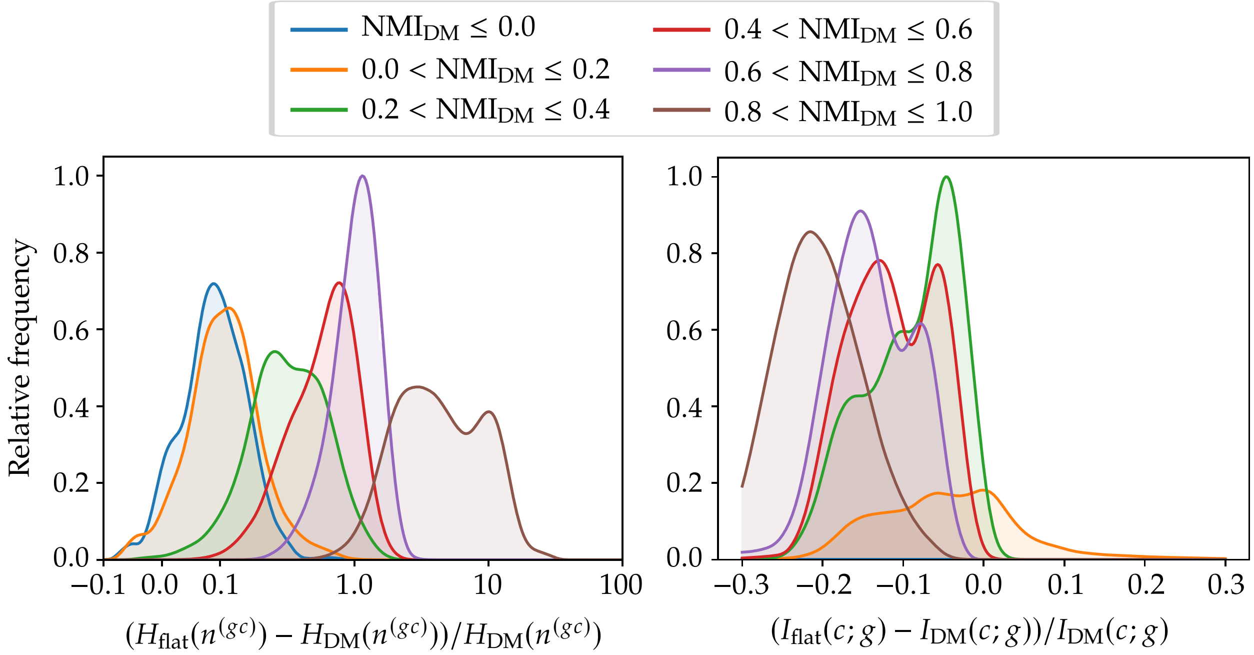
Based on these results we observe that when c and g are similar (NMI > 0.8, brown curve in the figure) the information gains when transmitting the contingency table are large, up to a factor of ten or more. This aligns with our observation, discussed in Appendix C.13, that for $c \simeq g$ the new encoding scheme is near-optimal, while the flat scheme is very inefficient. Even in cases where c and g are less similar, efficiency gains are often significant, typically above 10% and as high as 100% or more. There are a handful of cases, all occurring when the candidate labeling and ground truth are very dissimilar (NMI < 0.2, blue in the figure), where the new encoding performs slightly worse than the standard one, as discussed in Section 3.1.2.3. However, given that mutual information measures are normally applied in cases where the two labelings are significantly similar, the evidence of Fig. 3.8 suggests that our new encoding should be preferred, often by a wide margin, in most practical community detection scenarios.

As a result of the changes in both $H(g)$ and $H(g|c)$, the value of the mutual information itself can also change significantly. Figure 3.8b shows the fractional change in the mutual information in our test set, with the different curves again showing the results for different ranges of similarity between the ground truth and the candidate division. Because the standard encoding usually overestimates the conditional information $H(g|c)$, it tends to underestimate the mutual information $I(c;g) = H(g) - H(g|c)$, although this bias is offset somewhat by the corresponding overestimate of the unconditional entropy $H(g)$. On balance, however, the standard encoding significantly underestimates the mutual information in many cases and there are substantial information savings to be had under the new encoding, with the mutual information changing by up to 20% or more in the common case where the two labelings are similar (NMI > 0.8, brown curve in the figure).

3.1.4 Conclusions

In this section we have presented an improved formulation of the mutual information between two labelings of the same set of objects. Our approach is in the spirit of the recently proposed reduced mutual information, and like that measure it addresses the bias towards an excessive number of groups present in traditional measures by taking full account of information costs including particularly the cost of the contingency table. Where our proposal differs from the standard reduced mutual information is in using a more efficient encoding for the contingency table. While all information theoretic measures are, in a sense, merely bounds on the true value, our formulation gives significantly tighter bounds in the common regime where the two labelings are similar to one another.

We have demonstrated our proposed measure with a number of examples and performed extensive tests on network community structures generated using the LFR benchmark model. In the latter context we find that the new encoding does produce considerable



(a)

(b)

Figure 3.8: (a) Fractional change in the information cost of transmitting the contingency table $n^{(gc)}$ using the Dirichlet-multinomial encoding compared to the flat encoding. The different curves show the distribution of values for different ranges of similarity between the ground-truth and candidate labelings, as measured by the (Dirichlet-multinomial) normalized mutual information. The horizontal scale is linear between −0.1 and 0.1 and logarithmic outside that range. (b) Fractional change in the mutual information from the improved encoding of the contingency table and group sizes. The horizontal axis is linear across the entire range and the different curves again indicate distributions for different ranges of normalized mutual information. Cases where both mutual informations give a result of 0, for example when $q_c = 1$, have been removed, since they yield a fractional change of 0/0.

savings in information cost and the resulting values for the mutual information differ from the standard reduced mutual information by up to 20% of the total value under commonly occurring conditions.

Looking ahead, the improved encoding we present for contingency tables could also be used in applications beyond the computation of the mutual information that is the focus of this chapter. In general, the Dirichlet-multinomial distribution that underlies our encoding provides a more informative prior than the standard uniform distribution for Bayesian analysis involving contingency tables [59]. The encoding presented here could thus be used to improve data compression performance of any model that requires the specification of a prior distribution over contingency tables, for example in the methods for clustering discrete data presented in [70, 71].

3.2 Normalized mutual information

The mutual information presented in the previous section computes the absolute shared information content of two labelings of the same set of objects. Yet, this absolute number of bits is difficult to interpret in isolation. The mutual information is therefore often normalized to help contextualize the result. In this section we discuss how the symmetric *normalized mutual information* (NMI) typically used for this purpose also introduces a bias to the measure, and how we can instead normalize the mutual information in an unbiased way. We then use this corrected measure to compare the performance of a basket of popular algorithms for network community detection and show that one's conclusions about which algorithm is best are significantly affected by the biases in the traditional mutual information.

3.2.1 Introduction

As discussed in the last section, mutual information is a useful tool to compare two different labelings of a set. The measure works by asking how efficiently we can describe one labeling if we already know the other [34]. Specifically, it measures how much less information it takes to communicate the first labeling if we know the second versus if we do not. In this section, we apply the mutual information to evaluate the performance of algorithms for network community detection [36]. One takes a network whose community structure is already known and applies a community detection algorithm to it to infer the communities. Then one uses mutual information to compare the output of the algorithm to the known correct communities. Algorithms that consistently achieve high mutual information scores are considered good.

In this assessment, we must be weary of biases in our metric. For example, the traditional mutual information contains a bias towards labelings with too many groups. As shown in Section 3.1, this effect can be remedied by "reducing" the mutual information to better reflect the true information costs of transmission [65]. In practice, another drawback of the mutual information arises when the measure is normalized, as it commonly is to improve interpretability. The most popular normalization scheme creates a measure that runs between zero and one by dividing the mutual information by the arithmetic mean of the entropies of the two labelings being compared [10], although one can also normalize by the minimum, maximum, or geometric mean of the entropies. As we demonstrate in this section, however, these normalizations introduce biases into the results by comparison with the unnormalized measure, because the normalization factor depends on the candidate labeling as well as the ground truth. This effect can be large enough to change scientific outcomes, and we provide examples of this phenomenon.

In order to avoid this latter bias, while still retaining the interpretability of a normalized mutual information measure, we favor normalizing by the entropy of the ground-truth labeling alone. This removes the source of bias but introduces an asymmetry in the normalization. At first sight this asymmetry may seem undesirable, and previous authors have gone to some lengths to avoid it. Here, however, we argue that it is not only justified but actually desirable, for several reasons. First, many of the classification problems we consider are unaffected by the asymmetry, since they involve the comparison of one or more candidate labelings against a single, unchanging ground truth. Moreover, by contrast with the multitude of possible symmetric normalizations, the asymmetric measure we propose is the unique way to normalize the mutual information without introducing biases due to the normalization itself.

More broadly, the asymmetric measure is more informative than the conventional symmetric one. Consider for instance the common situation where the groups or communities in one labeling are a subdivision of those in the other. We might for example label individuals by the country they live in on the one hand and by the town or city on the other. Then the more detailed labeling tells us everything there is to know about the coarser one, but the reverse is not true: telling you the city fixes the country, but not *vice versa*. Thus one could argue that the mutual information between the two *should* be asymmetric, and this type of asymmetry will be a key feature of the measures we study.

Both drawbacks of the standard mutual information—bias towards too many groups and dependence of the normalization on the candidate labeling—can be addressed simultaneously by using an asymmetric normalized reduced mutual information as defined in this section. In support of this approach we present an extensive comparison of the performance of this and other variants of the mutual information in network community detection tasks, generating a large number of random test networks with known community structure and a variety of structural parameters, and then attempting to recover the communities using popular community detection algorithms. Within this framework, we find that conclusions about which algorithms perform best are significantly impacted by the choice of mutual information measure, and specifically that traditional measures erroneously favor algorithms that find too many communities, but our proposed measure does not.

3.2.2 Normalization of the mutual information

The absolute mutual information of two labelings g and c is equal to the total entropy of g minus the conditional entropy of g given c:

$$I(c;g) = H(g) - H(g|c). \tag{3.36}$$

Loosely, we say that $I(c;g)$ measures "how much c tells us about g." Since the conditional information content $H(g|c)$ is nonnegative, the mutual information must satisfy inequalities

$$I(c;g) \leq H(g), H(c), \tag{3.37}$$

which are saturated by

$$I(g;g) = H(g), \qquad I(c;c) = H(c). \tag{3.38}$$

In this work we consider three versions of the mutual information which each satisfy these inequalities. First, the traditional mutual information (MI) of Eq. (3.12),

$$I_0(c;g) = \log \frac{n! \prod_{rs} n_{rs}^{(gc)}!}{\prod_r n_r^{(g)}! \prod_s n_s^{(c)}!}, \tag{3.39}$$

a measure biased towards labelings c with too many groups. We consider two variants that avoid this bias: a correction for chance known as the adjusted mutual information (AMI),

$$I_A(c;g) = I_0(c;g) - \left\langle I_0(c;g) \right\rangle_{\{c|n^{(c)}\}}, \tag{3.40}$$

and the reduced mutual information (RMI) defined in Section 3.1,

$$\begin{aligned}
I(c;g) = \log \frac{n! \prod_{rs} n_{rs}^{(gc)}!}{\prod_r n_r^{(c)}! \prod_s n_s^{(g)}!} &+ \log \binom{n + q_g \alpha_g - 1}{q_g \alpha_g - 1} \\
&- \sum_{r=1}^{q_g} \log \binom{n_r^{(g)} + \alpha_g - 1}{\alpha_g - 1} \\
&- \sum_{s=1}^{q_c} \log \binom{n_s^{(c)} + q_g \alpha_{g|c} - 1}{q_g \alpha_{g|c} - 1} \\
&+ \sum_{r=1}^{q_g} \sum_{s=1}^{q_c} \log \binom{n_{rs}^{(gc)} + \alpha_{g|c} - 1}{\alpha_{g|c} + 1}. \qquad (3.41)
\end{aligned}$$

As discussed in Appendix C.18, although we prefer the reduced mutual information, both the AMI and RMI serve a similar role.

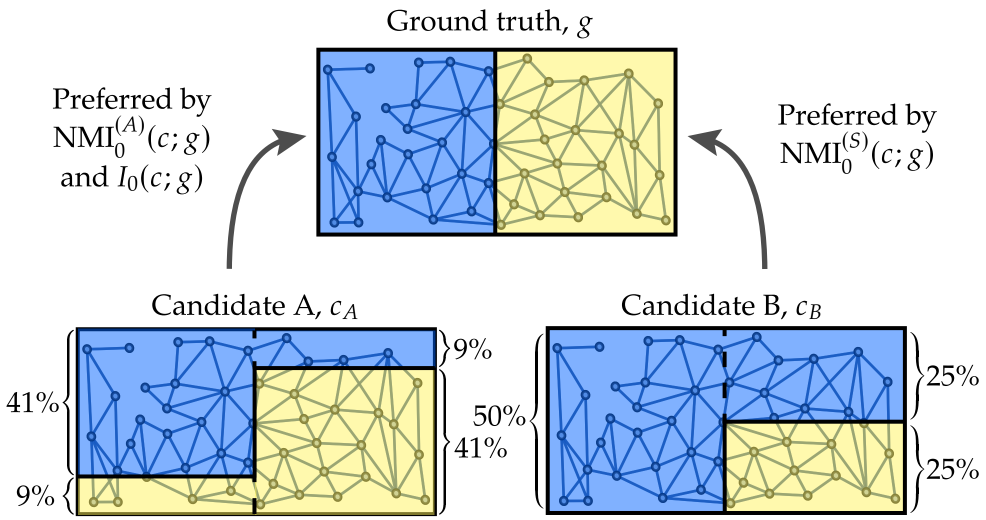
Regardless of our choice, a fundamental difficulty with mutual information as a measure of similarity is that its range of values depends on the particular application, which makes it difficult to say when a value is large or small. Is a mutual information of 10 a large value? Sometimes it is and sometimes it isn't, depending on the context. To get around this obstacle one commonly normalizes the mutual information so that it takes a maximum value of 1 when the candidate labeling agrees exactly with the ground truth. There are a number of ways this can be achieved and, as we show here, they are not all equal. In particular, some, including the most popularly used normalization, can result in biased results and should, in our opinion, be avoided. In its place, we propose an alternative, unbiased normalized measure.

The most common normalized measure, commonly referred to simply as the "normalized mutual information," uses the plain mutual information $I_0(c;g)$ as a base measure and normalizes it thus:

$$\mathrm{NMI}_0^{(S)}(c;g) = \frac{I_0(c;g)}{\frac{1}{2}[H_0(c) + H_0(g)]} = \frac{I_0(c;g) + I_0(g;c)}{I_0(c;c) + I_0(g;g)}. \qquad (3.42)$$

This measure has a number of desirable features. Because of the inequalities in (3.37), its value falls strictly between zero and one. And since both the base measure and the normalization are symmetric under interchange of c and g, the normalized measure also retains this symmetry (hence the superscript "(S)," for symmetric).

Equation (3.42) is not the only normalization that achieves these goals. Equation (3.37)


Ground truth, g
Preferred by $\mathrm{NMI}_0^{(A)}(c;g)$ and $I_0(c;g)$
Preferred by $\mathrm{NMI}_0^{(S)}(c;g)$
Candidate A, c_A
Candidate B, c_B
41%
9%
9%
41%
50%
25%
25%

Measure	c_A		c_B
$I_0(c;g)$	0.222	>	0.216
$\mathrm{NMI}_0^{(A)}(c;g)$	0.320	>	0.311
$\mathrm{NMI}_0^{(S)}(c;g)$	0.320	<	0.344

Figure 3.9: Normalization can impact which labeling is preferred by a mutual information measure. For the ground-truth labeling g (top), the standard unnormalized mutual information $I_0(c,g)$ of Eq. (3.39) (Stirling approximated) gives different scores to the two candidate labelings c_A and c_B, the former receiving a higher score than the latter. By definition, the asymmetrically normalized mutual information will always agree with the unnormalized measure, but the symmetrically normalized measure may not, as is the case here: in this example the symmetric measure scores candidate c_B higher. Note that the network itself plays no role here—it is included only as a visual aid, as the mutual information values depend only on the sizes of the shaded regions.

implies that

$$\begin{aligned} I_0(c;g) &\leq \min(I_0(c;c), I_0(g;g)) \\ &\leq \sqrt{I_0(c;c) I_0(g;g)} \\ &\leq \max(I_0(c;c), I_0(g;g)), \end{aligned} \tag{3.43}$$

which gives us three more options for a symmetric denominator in the normalized measure. The arithmetic mean in Eq. (3.42), however, sees the most use by far [75, 102, 140, 54].

We can extend the notion of symmetric normalization to any other base measure of mutual information $I_X(c;g)$ satisfying the inequality Eq. (3.37), such as adjusted or reduced mutual information [64], by writing

$$\mathrm{NMI}_X^{(S)}(c;g) = \frac{I_X(c;g) + I_X(g;c)}{I_X(c;c) + I_X(g;g)}. \tag{3.44}$$

All such measures, however, including the standard measure of Eq. (3.42), share a crucial shortcoming, that the normalization depends on the candidate labeling c and hence that the normalized measure can prefer a different candidate labeling to the base measure purely because of the normalization. Figure 3.9 shows an example of how this can occur. Two candidate labelings c_A and c_B are considered for the same ground truth g. Under the unnormalized mutual information of Eq. (3.39), candidate A receives a higher score than candidate B, but under the normalized measure of Eq. (3.42) the reverse is true. This behavior is due to the difference in entropy $I_0(c;c)$ between the two candidate divisions, the one on the right having larger entropy than the one on the left, which increases its normalization factor and correspondingly decreases the normalized mutual information.

We argue that the unnormalized measure is more correct on this question, having a direct justification in terms of information theory. The purpose of the normalization is purely to map the values of the measure onto a convenient numerical interval, and should not change outcomes as it does here. Moreover, different symmetric normalizations can produce different results. For instance, if one normalizes by $\max(I_0(c;c), I_0(g;g))$ in Fig. 3.9 then candidate c_A is favored in all cases.

These issues are unavoidable when using a symmetric normalization scheme. In any such scheme the normalization must depend on both c and g and hence can vary with the candidate labeling. However, if we drop the requirement of symmetry then we can normalize in a way that avoids these issues. We define the *asymmetric normalization* of any base measure I_X as

$$\mathrm{NMI}_X^{(A)}(c;g) = \frac{I_X(c;g)}{I_X(g;g)}. \tag{3.45}$$

This definition still gives $\mathrm{NMI}_X^{(A)}(g;g) = 1$, but now the normalization factor in the denominator has no effect on choices between candidate labelings, since it is independent of c. In fact, Eq. (3.45) is the only way to normalize such that $I_X^{(A)}(g;g) = 1$ while simultaneously ensuring that the preferred candidate is always the same as for the base measure. Thus this measure also removes any ambiguity about how one should perform the normalization. Loosely, this asymmetrically normalized mutual information measures "how much c tells us about g as a fraction of all there is to know about g." Such asymmetric choices of normalization have appeared in other contexts under the name "coefficient of constraint" or "coefficient of uncertainty" [33, 126, 49], although symmetric normalizations are almost universally used in the model validation context, where they introduce the bias we discuss here.

The amount of bias inherent in the symmetrically normalized measure when compared with the asymmetric one can be quantified by the ratio between the two:

$$\begin{aligned}\frac{\mathrm{NMI}_X^{(S)}(c;g)}{\mathrm{NMI}_X^{(A)}(c;g)} &= \frac{[I_X(c;g) + I_X(g;c)]/[I_X(c;c) + I_X(g;g)]}{I_X(c;g)/I_X(g;g)} \\ &= \frac{1 + I_X(g;c)/I_X(c;g)}{1 + I_X(c;c)/I_X(g;g)}.\end{aligned} \tag{3.46}$$

If the base measure I_X is itself symmetric (which it is, either exactly or approximately, for all the measures we consider), then this simplifies further to

$$\frac{\mathrm{NMI}_X^{(S)}(c;g)}{\mathrm{NMI}_X^{(A)}(c;g)} = \frac{H(g)}{\frac{1}{2}[H(g) + H(c)]}. \tag{3.47}$$

Values of this quantity below (above) 1 indicate that the value of the symmetric measure is biased low (high). Thus, for instance, complex candidate labelings c that have higher entropy than the ground truth will result in a symmetric measure whose values are too low. As discussed in the introduction, traditional mutual information measures are particularly problematic when the candidate is a *refinement* of the ground truth, meaning the candidate groups are subsets of the ground-truth groups. In that case the traditional measure returns values that are too high. Equation (3.47) implies that, to some extent, the symmetric normalization will correct this issue: a candidate c that takes the form of a refinement of g will have $H(c) > H(g)$, which will lower the value of the symmetrically normalized mutual information. This may in part explain why these biases have been overlooked in the past: two wrongs have conveniently canceled to make a right.

We argue, however, that this is not the best way to address the problem and that the correct approach is instead to use the reduced mutual information. The reduced mutual information also corrects for the case where one labeling is a refinement of the other, but does so in a more principled manner that directly addresses the root cause of the problem, rather than merely penalizing complex candidate labelings in an *ad hoc* manner as a side-effect of normalization.

We will see examples of these effects in Section 3.2.3, where we apply the various measures to community detection in networks and find that indeed the traditional symmetrically normalized mutual information is biased. In some cases it is fortuitously biased in the right direction, although it is still problematic in some others.

An obvious downside of asymmetric normalization is the loss of the symmetry in the final measure. In the most common applications of normalized mutual information, where labelings are evaluated against a ground truth, an inherently asymmetric situation, the asymmetric measure makes sense, but in other cases the lack of symmetry can be undesirable. Embedding and visualization methods that employ mutual information as a similarity measure, for example, normally demand symmetry [104]. And in cases where one is comparing two candidate labelings directly to one another, rather than to a separate ground truth, a symmetric measure may be preferable. Even in this latter case, however, the asymmetric measure may sometimes be the better choice, as discussed in the introduction. For instance, when one labeling c_1 is a refinement of the other c_2, the information content is inherently asymmetric: c_1 says more about c_2 than c_2 does about c_1. An explicit example of this type of asymmetry is shown in Figure 3.10, where we consider two labelings of 27 objects. The left labeling c_1 is a detailed partition of the objects into nine small groups while the right labeling c_2 is a coarser partition into only three groups, each of which is an amalgamation of three of the smaller groups in c_1. Because of this nested relationship, it is relatively easy to transmit c_2 given knowledge of c_1 but more difficult to do the reverse. This imbalance is reflected in the asymmetric normalized mutual information values in each direction (top and bottom arrows in the figure), but absent from the symmetric version (middle row).

Combining the benefits of asymmetric normalization and the reduced mutual information, we advocate in favor of the asymmetrically-normalized reduced mutual information defined by

$$\mathrm{NMI}^{(A)}(c;g) = \frac{I(c;g)}{I(g;g)}, \tag{3.48}$$

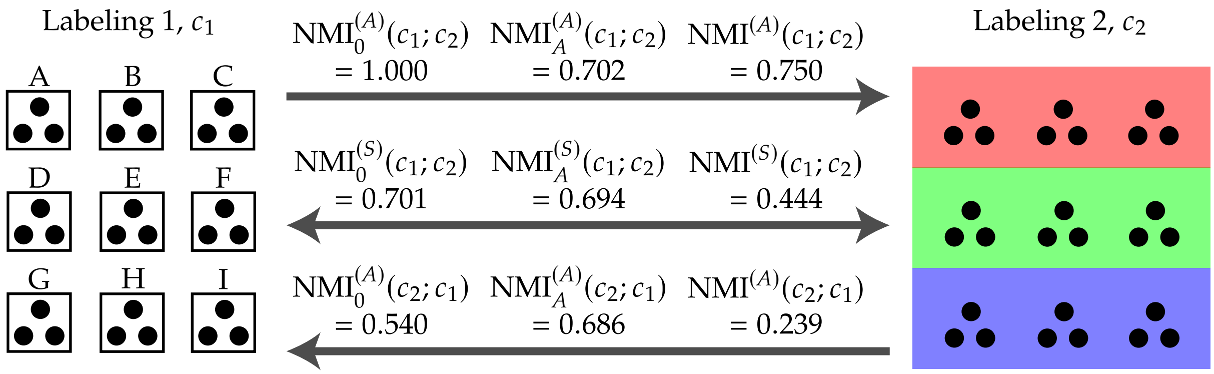

Labeling 1, c_1
A
B
C
D
E
F
G
H
I
$\mathrm{NMI}_0^{(A)}(c_1;c_2)$ = 1.000
$\mathrm{NMI}_A^{(A)}(c_1;c_2)$ = 0.702
$\mathrm{NMI}^{(A)}(c_1;c_2)$ = 0.750
$\mathrm{NMI}_0^{(S)}(c_1;c_2)$ = 0.701
$\mathrm{NMI}_A^{(S)}(c_1;c_2)$ = 0.694
$\mathrm{NMI}^{(S)}(c_1;c_2)$ = 0.444
$\mathrm{NMI}_0^{(A)}(c_2;c_1)$ = 0.540
$\mathrm{NMI}_A^{(A)}(c_2;c_1)$ = 0.686
$\mathrm{NMI}^{(A)}(c_2;c_1)$ = 0.239
Labeling 2, c_2

Figure 3.10: Values of the various normalized mutual information measures between two example partitions of the same set of 27 objects. The left partition c_1 divides the objects into nine groups of three objects each and is a refinement of the right partition c_2, which divides them into only three groups. The arrows indicate the direction of the comparison and the accompanying notations give the values of the mutual information measures. All three asymmetrically normalized measures capture the intuition that the partition c_1 tells us more about c_2 than c_2 tells us about c_1.

where the mutual information $I(c;g)$ is quantified as in Eq. (3.41). This measure correctly accounts for the information contained in the contingency table, returns a negative value when c is unhelpful for recovering the ground truth, returns 1 if and only if $c = g$, and always favors the same labeling as the unnormalized measure. The traditional normalized mutual information possesses none of these desirable qualities.

3.2.3 Example application: Community detection

As an example of the performance of the various measures discussed above, we revisit the extensive series of tests of algorithms for network community detection used in Section 3.1. In these examples we apply the LFR graph model [77] to generate random test networks with known community structure and realistic distributions of node degrees and group sizes, and then attempt to recover that structure using a variety of standard community detection algorithms, quantifying the accuracy of the recovery with six different measures: symmetrically and asymmetrically normalized versions of the traditional mutual information, the adjusted mutual information, and the reduced mutual information. A number of studies have been performed in the past to test the efficacy of community detection algorithms on LFR benchmark networks [75, 102, 140, 54], but using only the symmetrically normalized, non-reduced mutual information as a similarity measure. Our results indicate that this measure can produce biased outcomes and we recommend the asymmetric reduced mutual information instead.

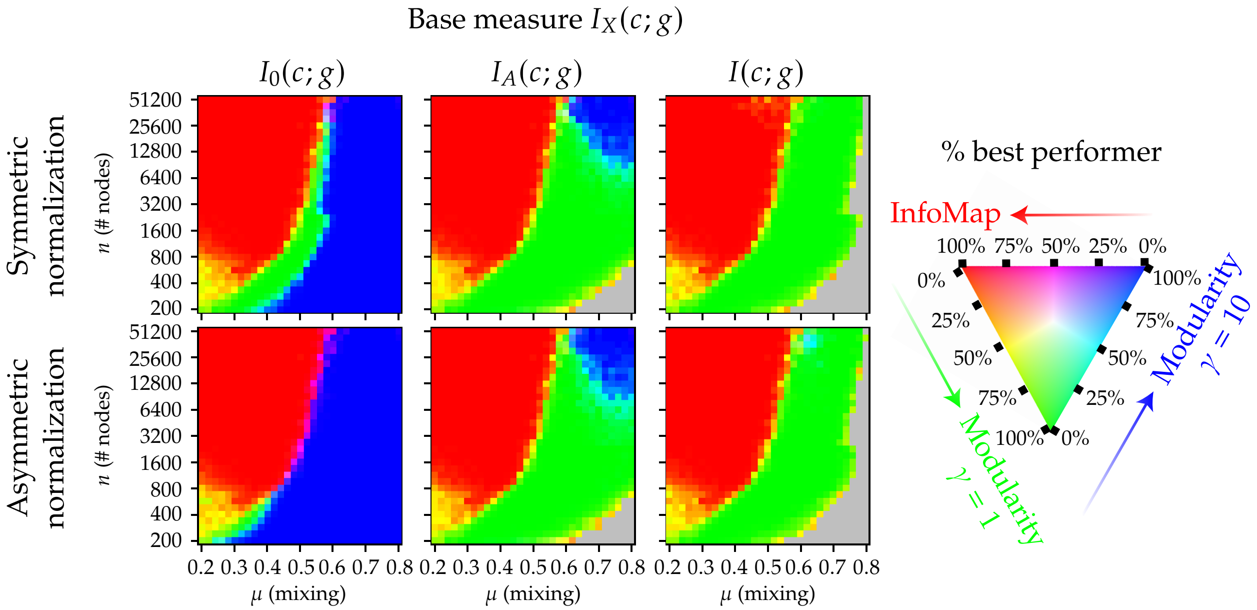


Figure 3.11: Performance of three community detection algorithms: InfoMap (red) and modularity maximization with $\gamma = 1$ (green) and $\gamma = 10$ (blue)—see key on right. The colors in each panel of this figure indicate which of the three is best able to find the known communities in a large set of LFR benchmark networks, according to the six mutual information measures we consider. Mixtures of red, green, and blue denote the proportions of test cases in which each algorithm performs best. Regions in gray indicate parameter values for which no algorithm achieved a positive mutual information score.

The LFR model contains a number of free parameters that control the size of the networks generated, their degree distribution, the distribution of community sizes, and the relative probability of within- and between-group edges. (See Appendix C.16 for details of the LFR generative process.) We find that the distributions of degrees and community sizes do not significantly impact the relative performance of the various algorithms tested and that performance differences are driven primarily by the size n of the networks and the mixing parameter μ that controls the ratio of connections within and between groups, so our tests focus on performance as a function of these parameters.

3.2.4 Comparison between variants of the mutual information

Figure 3.11 summarizes the relative performance of the various mutual information measures in our tests. In this set of tests we limit ourselves, for the sake of clarity, to the top three community detection algorithms—InfoMap and the two variants of modularity maximization—and measure which of the three returns the best results according to each of our six mutual information measures, as a function of network size n and the mixing parameter μ. Each point in each of the six panels is color-coded with some mix of red, green, and blue to indicate in what fraction of cases each of the algorithms performs best according to each of the six measures and, as we can see, the results vary significantly among measures. An experimenter trying to choose the best algorithm would come to substantially different conclusions depending on which measure they use.

One consistent feature of all six mutual information measures is the large red area in each panel of Fig. 3.11, which represents the region in which the InfoMap algorithm performs best. Regardless of the measure used, InfoMap is the best performer on networks with low mixing parameter (i.e., strong community structure) and relatively large network size. For higher mixing (weaker structure) or smaller network sizes, modularity maximization does better. Which version of modularity is best, however, depends strongly on the mutual information measure. The traditional symmetrically normalized mutual information (top left panel) mostly favors the version with a high resolution parameter of $\gamma = 10$ (blue), but the asymmetric reduced measure for which we advocate (bottom right) favors the version with $\gamma = 1$ (green). (The regions colored gray in the figure are those in which no algorithm receives a positive mutual information score and hence all algorithms can be interpreted as failing.)

These results raise significant doubts about the traditional measure. Consider the lower right corner of each plot in Fig. 3.11, which is the regime of small network sizes n and high mixing μ so that the community signal is weak and the noise is high. Here, the adjusted and reduced mutual informations indicate that all algorithms are failing. This is expected: for very weak community structure all detection algorithms are expected to show a "detectability threshold" beyond which they are unable to identify any communities [42, 86, 93]. The standard normalized mutual information, on the other hand, claims to find community structure in this regime using the $\gamma = 10$ version of modularity maximization. This occurs because the $\gamma = 10$ algorithm finds many small communities and, as discussed in Section 3.1.2, a labeling with many communities, even completely random ones, is accorded a high score by a non-reduced mutual information. The presence of this effect is demonstrated explicitly in Appendix C.16.1 for the case of modularity maximization with $\gamma = 10$, and this offers a clear reason to avoid the standard measure.

The bottom left panel in Fig. 3.11 shows results for the asymmetrically normalized version of the traditional mutual information, which gives even worse results than the symmetric version, with hardly any region in which the $\gamma = 1$ version of modularity maximization outperforms the $\gamma = 10$ version. This behavior arises for the reasons discussed in Section 3.2.2—the bias inherent in the symmetric normalization fortuitously acts to partially correct the errors introduced by neglecting the information content of the

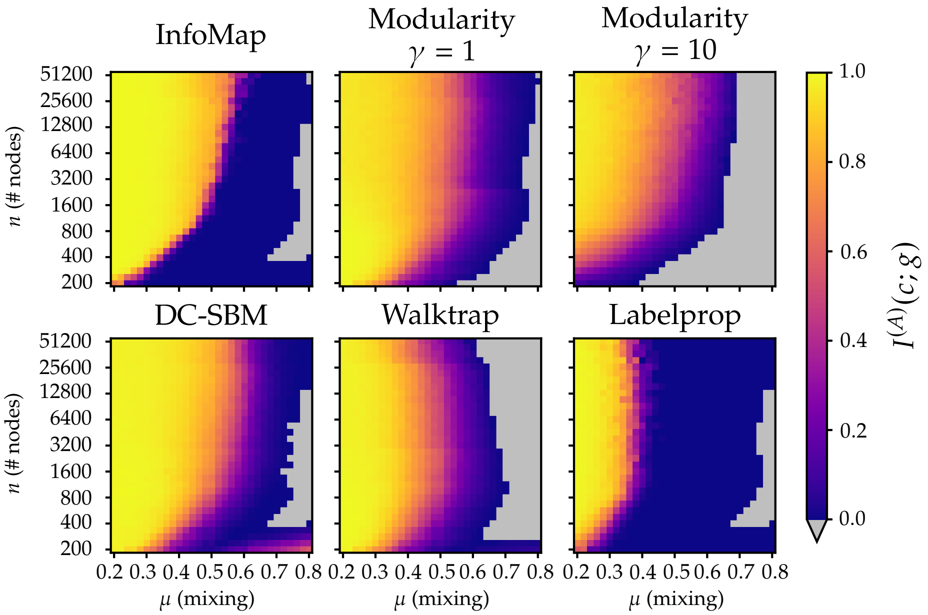


Figure 3.12: Performance of each of the six community detection algorithms considered here in identifying the known communities in a large set of LFR benchmark networks, as quantified by the asymmetrically normalized reduced information measure $I^{(A)}(c;g)$ proposed in this work. Gray areas indicate parameter values for which NMI < 0, so that the conditional encoding is less efficient than the direct encoding.

contingency table. The asymmetric normalization eliminates this correction and hence performs more poorly. The correct solution to this problem, however, is not to use a symmetric normalization, which can bias outcomes in other ways as we have seen, but rather to adopt a reduced mutual information measure.

Finally, comparing the middle and right-hand columns of Fig. 3.11, we see that the results for the adjusted and reduced mutual information measures are quite similar in these tests, although there are some differences. In particular, the adjusted measure appears to find more significant structure for higher mixing than the reduced measure. This occurs because, as discussed in Appendix C.18 and Ref. [101], the adjusted measure encodes the contingency table in a way that is optimized for more uniform tables than the reduced measure, and thus penalizes uniform tables less severely, leading to overestimates of the mutual information in the regime where detection fails—in this regime the candidate and ground-truth labelings are uncorrelated which results in a uniform contingency table. This provides further evidence in favor of using a reduced mutual information measure.

3.2.5 Comparison between community detection algorithms

Settling on the asymmetrically normalized reduced mutual information as our preferred measure of similarity, we now ask which community detection algorithm or algorithms perform best according to this measure? We have already given away the answer—InfoMap and modularity maximization get the nod—but here we give evidence for that conclusion.

Figure 3.12 shows results for all six algorithms listed in Section 3.2.3. Examining the figure, we see that in general the best-performing methods are InfoMap, traditional modularity maximization with $\gamma = 1$, and the inference method using the degree-corrected stochastic block model[1]. Among the algorithms considered, InfoMap achieves the highest mutual information scores for lower values of the mixing parameter μ in the LFR model, but fails abruptly as μ increases, so that beyond a fairly sharp cutoff around $\mu = 0.5$ other algorithms do better. As noted by previous authors [11], InfoMap's specific failure mode is that it places all nodes in a single community and this behavior can be used as a simple indicator of the failure regime. In this regime one must use another algorithm. Either the modularity or inference method are reasonable options, but modularity has a slight edge, except in a thin band of intermediate μ values which, in the interests of simplicity, we choose to ignore. (We discuss some caveats regarding the relationship between the degree-corrected stochastic block model and the LFR benchmark in Appendix C.16.)

Thus—always assuming the LFR benchmark is a good test of performance—our recommendations for the best community detection algorithm are relatively straightforward. If we are in a regime where InfoMap succeeds, meaning it finds more than one community, then one should use InfoMap. If not, one should use standard modularity maximization with $\gamma = 1$. That still leaves open the question of how the modularity should be maximized. In our studies we find the best results with simulated annealing, but simulated annealing is computationally expensive. In regimes where it is not feasible, we recommend using the Leiden algorithm instead. (Tests using other computationally efficient maximization schemes, such as the Louvain and spectral algorithms, generally performed less well than the Leiden algorithm.)

3.2.6 Conclusions

In this chapter we have examined the performance of a range of mutual information measures for comparing labelings of objects in classification, clustering, or community

[1] The degree-corrected stochastic block model used in these tests is not quite the same as the version described in Section 2.1. See Appendix C.16.2 for details.

detection applications. We argue that the commonly used normalized mutual information is biased in two ways: (1) because it ignores the information content of the contingency table, which can be large, and (2) because the symmetric normalization it employs introduces spurious dependence on the labeling. We argue in favor of a different measure, an asymmetrically normalized version of the reduced mutual information, which rectifies both of these shortcomings.

To demonstrate the effects of using different mutual information measures, we have presented results of an extensive set of numerical tests on popular network community detection algorithms, as evaluated by the various measures we consider. We find that conclusions about which algorithms are best depend substantially on which measure we use.

Chapter 4

Hierarchies

In Chapters 2 and 3, we discussed how to find and evaluate group structures in undirected networks. In this chapter, we focus instead on the *directed* structure of networks, and the hierarchies they reveal.

We reformulate and extend the Bradley-Terry model described in Section 1.2.4 to model two features: an element of randomness or luck that leads to upset wins, and a "depth of competition" variable that measures the complexity of a game or hierarchy. Fitting the resulting model we estimate depth and luck in a range of games, sports, and social situations. In general, we find that social competition tends to be "deep," meaning it has a pronounced hierarchy with many distinct levels, but also that there is often a nonzero chance of an upset victory. Competition in sports and games, by contrast, tends to be shallow and in most cases there is little evidence of upset wins. Our presentation here is based upon [67], work in collaboration with Mark Newman.

4.1 Introduction

One of the oldest and best-studied problems in data science is the ranking of a set of items, individuals, or teams based on the results of pairwise comparisons between them. Such problems arise in sports, games, and other competitive human interactions, in paired comparison surveys in market research and consumer choice, in revealed-preference studies of human behavior, and in studies of social hierarchies in both humans and animals. In each of these settings, exemplified in Section 1.1.3, one has a set of comparisons between pairs of items or competitors, with outcomes of the form "A beats B" or "A is preferred to B," and the goal is to determine a ranking of competitors from best to worst, allowing for the fact that the data may be sparse (there may be no data for many pairs) or contradictory (e.g., A beats B beats C beats A). A group of chess players might play in a tournament, for example, and record wins and losses against each other. Consumers might express preferences between pairs of competing products, either directly in a survey or implicitly through their purchases or other actions. A flock of chickens might peck each other as a researcher records who pecked whom in order to establish the classic "pecking order."

A large number of methods have been proposed for solving ranking problems of this kind—see Refs. [37, 25, 78] for reviews. In this chapter we consider one of the most common, which uses a statistical model for wins and losses and then fits that model to observed win/loss data. In the most widely adopted version one considers a population of n competitors labeled by $i = 1 \ldots n$ and assigns to each a real score parameter $s_i \in [-\infty, \infty]$. Then the probability that i beats j in a single pairwise match or contest is assumed to be some function of the difference of their scores: $p_{ij} = f(s_i - s_j)$. The score function $f(s)$ satisfies the following axioms:

1. It is increasing in s, since by definition a better competitor has a higher probability of winning than a worse one.

2. It tends to 1 as $s \to \infty$ and to 0 as $s \to -\infty$, meaning that an infinitely good player always wins and an infinitely poor one always loses.

3. It is antisymmetric about its mid-point at $s = 0$, with the form

$$f(-s) = 1 - f(s), \tag{4.1}$$

   because the probability of losing is one minus the probability of winning. As a corollary, this also implies that the probability $f(0)$ of beating an evenly matched opponent is always $\frac{1}{2}$.

Subject to these constraints the function can still take a wide variety of forms, but the most popular choice by far is the logistic function $f(s) = 1/(1 + e^{-s})$—shown as the bold curve in Fig. 4.1a—which gives

$$f(s_i - s_j) = \frac{e^{s_i}}{e^{s_i} + e^{s_j}}. \tag{4.2}$$

This score function yields the Bradley-Terry model familiar from Section 1.2.4.

Given the model, one can estimate the values of the score parameters s_i by a number of standard methods, including maximum likelihood estimation [143, 20, 52, 63, 100], maximum a posteriori estimation [136], or Bayesian methods [38, 24], then rank competitors from best to worst in order of their scores. The fitted model can also be used to predict the outcome of future contests between any pair of competitors, even if they have never directly competed in the past.

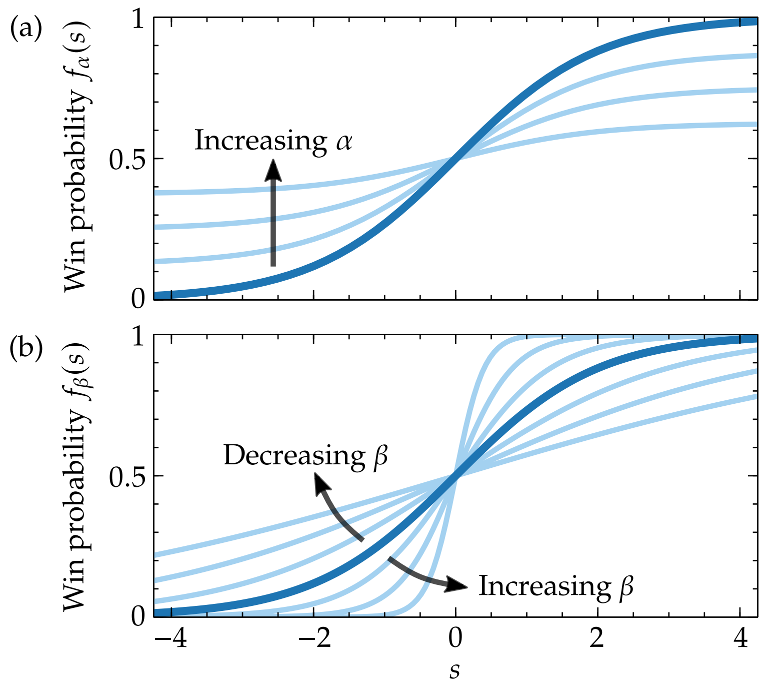


Figure 4.1: Score functions $f(s)$. (a) The bold curve represents the standard logistic function $f(s) = 1/(1 + e^{-s})$ used in the Bradley-Terry model. The remaining curves show the function f_α of Eq. (4.8) for increasing values of the luck parameter α. (b) The score function f_β of Eq. (4.9) for different values of the depth of competition β, both greater than 1 (steeper) and less than 1 (shallower).

This approach is effective and widely used, but the standard Bradley-Terry model is a simplistic representation of the patterns of actual competition and omits many important elements found in real-world interactions. Generalizations of the model have been proposed that incorporate some of these elements, such as the possibility of ties or draws between competitors [114, 39], multiway competition as in a horse race [83, 111], the "home-field advantage" of playing on your own turf [8], or multidimensional score parameters that allow for intransitive win probabilities between competitors [28, 85]. In this chapter we consider a further extension of the model that incorporates two additional features of particular interest, which have received comparatively little previous attention: the element of luck inherent for instance in games of chance, and the notion of "depth of competition," which captures the complexity of games or the number of distinct levels in a social hierarchy. In the remainder of the chapter we define and motivate this model and then describe a Bayesian approach for fitting it to data, which we use to infer the values of the luck and depth variables for a variety of real-world data sets drawn from different arenas of human and animal competition. Our results suggest that social hierarchies are in general deeper and may have a larger element of luck to their dynamics than recreational games and sports, which tend to be shallower and show little evidence of a luck component.

Software implementations of the various methods described in this chapter are available at https://github.com/maxjerdee/pairwise-ranking and https://doi.org/10.5061/dryad.kh18932fc.

4.2 The model

Suppose we observe m matches between n players. The outcomes of the matches can be represented by an $n \times n$ matrix $\boldsymbol{A}$ with element A_{ij} equal to the number of times player i beats player j. Within the standard Bradley-Terry model the probability of a win is given by Eq. (4.2) and, assuming the matches to be statistically independent, the probability or likelihood of the complete set of observed outcomes is

$$P(\boldsymbol{A}|\boldsymbol{s}) = \prod_{ij} f(s_i - s_j)^{A_{ij}} = \prod_{ij} \left(\frac{e^{s_i}}{e^{s_i} + e^{s_j}} \right)^{A_{ij}}, \tag{4.3}$$

where $\boldsymbol{s}$ is the vector with elements s_i and terms that only depend on the data $\boldsymbol{A}$ have been dropped. (We assume that the structure of the tournament—who plays whom—is determined separately, so that (4.3) is a distribution over the directions of the wins and losses only and not over which pairs of players competed.)

The scores are traditionally estimated by the method of maximum likelihood, maximizing (4.3) with respect to all s_i simultaneously to give estimates

$$\hat{\boldsymbol{s}} = \text{argmax}_{\boldsymbol{s}} P(\boldsymbol{A}|\boldsymbol{s}). \tag{4.4}$$

These maximum likelihood estimates (MLEs) can then be sorted in order to give a ranking of the competitors, or simply reported as measures of strength in their own right. The widely used Elo ranking system for chess players, for example, is essentially a version of this approach, but extended to allow for dynamic updates as new matches are added to the data set.

The maximum likelihood approach unfortunately has some drawbacks. For one, the likelihood is invariant under a uniform additive shift of all scores s_i and hence the scores are not strictly identifiable, though this issue can easily be fixed by normalization. A more serious problem is that the likelihood maximum does not exist at all unless the network of interactions—the directed network with adjacency matrix $\boldsymbol{A}$—is strongly connected (meaning there is a directed chain of victories from any player to any other), and the maximum likelihood estimation procedure fails, with the divergence of some or all of the scores, unless this relatively stringent condition is met.

This issue can be addressed by introducing a prior on the scores and adopting a Bayesian perspective. A variety of potential priors for this purpose have been systematically examined by Whelan [136], who, after careful consideration, recommends a Gaussian prior with mean zero. The variance is arbitrary—it merely sets the scale on which the score s is measured—but for subsequent convenience we here choose a variance of $\frac{1}{2}$ so that the prior on $\boldsymbol{s}$ takes the form

$$P(\boldsymbol{s}) = \prod_{i=1}^{n} \frac{1}{\sqrt{\pi}} e^{-s_i^2}. \tag{4.5}$$

An alternative prior, also recommended by Whelan, is the logistic distribution

$$P_L(\boldsymbol{s}) = \prod_{i=1}^{n} \frac{1}{(1+e^{s_i})(1+e^{-s_i})}. \tag{4.6}$$

In practice the Gaussian and logistic distributions are similar in shape and the choice of one or the other does not make a great deal of difference. The logistic distribution is perhaps the less natural of the two and we primarily use the Gaussian distribution in this chapter, but the logistic distribution does have the advantage of leading to faster numerical algorithms and we have used it in previous work for this reason [99, 100]. We also include it in the basket of models that we compare in Section 4.5.

Once we have defined a prior on the scores we can calculate a maximum a posteriori (MAP) estimate of their values as

$$\hat{\boldsymbol{s}} = \text{argmax}_{\boldsymbol{s}} P(\boldsymbol{s}|\boldsymbol{A}) = \text{argmax}_{\boldsymbol{s}} P(\boldsymbol{A}|\boldsymbol{s}) P(\boldsymbol{s}). \tag{4.7}$$

The MAP estimate always exists regardless of whether the interaction network is strongly connected, and using a prior also eliminates the invariance of the probability under an additive shift and hence the need for normalization. As an alternative to computing a MAP estimate we can also simply return the full posterior distribution $P(\boldsymbol{s}|\boldsymbol{A})$, which gives us complete information on the expected values and uncertainty of the scores given the observed data.

4.3 Extensions of the model

In this section we define generalizations of the Bradley-Terry model that extend the score function f in two useful ways, while keeping other aspects of the model fixed, including the normal prior. The specific generalizations we consider involve dilation or contraction of the score function in the vertical and horizontal directions. Vertical variation controls the element of luck that allows a weak player to sometimes beat a strong one; horizontal variation controls the "depth of competition," a measure of the complexity of a game or contest.

4.3.1 Upset wins and luck

The first generalization of the Bradley-Terry model that we consider is one where the function f is contracted in the vertical direction, as shown in Fig. 4.1a. We parametrize this function in the form

$$f_\alpha(s) = \tfrac{1}{2}\alpha + (1-\alpha)\frac{1}{1+e^{-s}}, \tag{4.8}$$

with $\alpha \in [0,1]$. In the traditional Bradley-Terry model $f(s)$ tends to 0 and 1 as $s \to \pm\infty$, as discussed in the introduction, but in the modified model with $\alpha > 0$ this is no longer the case. One can think of the parameter α as controlling the probability of an "upset win" in which an infinitely good player loses or an infinitely bad player wins. (The probabilities of these two events must be the same because of the antisymmetry condition, Eq. (4.1).)

For some games or competitions it is reasonable that $f(s)$ tends to zero and one at the limits. In a game like chess that has no element of randomness, an infinitely good player may indeed win every time. In a game of pure luck like roulette, on the other hand, both players have equal probability $\frac{1}{2}$ of winning, regardless of skill. These two cases correspond to the extreme values $\alpha = 0$ and $\alpha = 1$ respectively in Eq. (4.8). Values in between represent games that combine both luck and skill, like poker or backgammon, with the precise value of α representing the proportion of luck. For this reason we refer to α as the luck parameter, or simply the "luck."

(One could also consider the chance of the weaker player winning in the standard Bradley-Terry model to be an example of luck or an upset win, but that is not how we use these words here. In the present context the "luck" α describes the probability of winning the game even if one's opponent is infinitely good, which is zero in the standard model but nonzero in the model of Eq. (4.8) with $\alpha > 0$.)

Another way to think about α is to imagine a game as a mixture of a luck portion and a skill portion. With probability α the players play a game of pure chance in which the winner is chosen at random, for instance by the toss of a coin. Alternatively, with probability $1 - \alpha$, they play a game of skill, such as chess, and the winner is chosen with the standard Bradley-Terry probability. The overall probability of winning is then given by Eq. (4.8) and the parameter α represents the fraction of time the game is decided by pure luck. By fitting (4.8) to observed win-loss data we can learn the luck inherent in a competition or hierarchy. We do this for a variety of data sets in Section 4.4.

4.3.2 Depth of competition

The second generalization we consider is one where the function f is dilated or contracted in the horizontal direction, as shown in Fig. 4.1b, by a uniform factor $\beta > 0$ thus:

$$f_\beta(s) = \frac{1}{1 + e^{-\beta s}}. \tag{4.9}$$

The slope of this function at $s = 0$ is given by

$$f'_\beta(0) = \left[\frac{\beta e^{-\beta s}}{(1 + e^{-\beta s})^2} \right]_{s=0} = \tfrac{1}{4}\beta, \tag{4.10}$$

so β is simply proportional to the slope. A more functional way of thinking about β is in terms of the probability that the stronger of a typical pair of competitors will win. With a normal prior on s of variance $\frac{1}{2}$ as described in Section 4.2, the difference $s_i - s_j$ between the scores of a randomly chosen pair of competitors will be *a priori* normally distributed with variance 1, meaning the scores will be separated by an average (root-mean-square) distance of 1. Consider two players separated by this average distance. If β is small, making f_β a relatively flat function (the shallowest curve in Fig. 4.1b), the probability p_{ij} of the stronger player winning will be close to $\frac{1}{2}$ and there is a substantial chance that the weaker player will win. Conversely, if β is large then p_{ij} will be close to 1 (the steepest curve in Fig. 4.1b) and the stronger player is very likely to prevail.

This horizontal dilation of the score function $f(s)$ for a fixed width prior $P(\boldsymbol{s})$ is exactly equivalent to instead dilating the prior $P(\boldsymbol{s})$ and keeping the score function fixed as described in Section 1.2.4 and illustrated in Figure 1.16. The parameter β in Eq. (4.9) therefore serves the same role as the *depth of competition* parameter β in Eq. (1.57) and controls the imbalance in strength or skill between the average pair of players. We will therefore refer to the parameter in either formulation as the depth. Compared to the presentation in Section 1.2.4, viewing the depth as a transformation of the score function allows it to interact more directly with the luck, a vertical dilation of the score function.

In either formulation, the behavioral definition of one "level" of skill is the same. In Section 1.2.4 we considered the win probability achieved by a skill difference $\Delta s = s_i - s_j = 1$ and unit logistic score function,

$$p_{ij} = \frac{1}{1 + e^{-\Delta s}} = \frac{1}{1 + e^{-1}} = 0.731... \tag{4.11}$$

If we instead dilate the score function by a factor β, we can still define a "level" of skill as the distance between scores $\Delta s = s_i - s_j$ such that i beats j with the same probability of about 73%,

$$p_{ij} = \frac{1}{1 + e^{-\beta \Delta s}} = \frac{1}{1 + e^{-1}}. \tag{4.12}$$

Therefore at a skill difference of $\Delta s = 1/\beta$ we observe the same win probability that defines a "level" of play. Since the form of the prior is fixed, there are therefore β such meaningful levels between the typical pair of players.

This count β can be thought of as a measure of the complexity or depth of a game or competition. A "deep" game, in this sense, is one that can be played at many levels, with players at each level markedly better than those at the level below. Chess, which is played at a wide range of skill levels from beginner to grandmaster, might be an example.

This concept of depth has a long history. For example, in an article in the trade publication *Inside Backgammon* in 1980 [117], world backgammon champion William Robertie defined a "skill differential" as the strength difference between two players that results in the better one winning 70 to 75% of the time—precisely our definition of a "level"—and the "complexity number" of a sport or game as the number of such skill differentials that separate the best player from the worst. Cauwet *et al.* [26] have defined a similar but more formal measure of game depth that they call "playing-level complexity." There has also been discussion in the animal behavior literature of the "steepness" of animal dominance hierarchies [41], which appears to correspond to roughly the same idea.

One should be careful about the details. Robertie and Cauwet *et al.* both define their measures in terms of the skill range between the best and worst players, but this could be problematic in that the range will depend on the particular sample of players one has and will tend to increase as the sample size gets larger, which seems undesirable. Our definition avoids this by considering not the best and worst players in a competition but the average pair of players, which gives a depth measure that is asymptotically independent of sample size.

Even when defined in this way, however, the number of levels is not solely about the intrinsic complexity of the game, but does also depend on who is competing. For example, if a certain competition is restricted to contestants who all fall in a narrow skill range, then β will be small even for a complex game. In a world-class chess tournament, for instance, where every player is an international master or better, the number of levels of play will be relatively small even though chess as a whole has many levels. Thus empirical values of β combine aspects of the complexity of the game with aspects of the competing population.

For this reason we avoid terms such as "complexity number" and "depth of game" that imply a focus on the game alone and refer to β instead as the "depth of competition," which we feel better reflects its meaning. (A variety of alternative notions of depth are discussed in Appendix D.22.)

4.3.3 Combined model

Combining both the luck and depth of competition variables into a single model gives us the score function

$$f_{\alpha\beta}(s) = \tfrac{1}{2}\alpha + (1-\alpha)\frac{1}{1+e^{-\beta s}}. \tag{4.13}$$

In Section 4.4, we fit this form to observed data from a range of different areas of study in order to infer the values of α and β. In the process one can also infer the scores s_i, which can be used to rank the participants or predict the outcome of unobserved contests, and we explore this angle later in the chapter. In this section, however, our primary focus is on α and β and on understanding the varying levels of luck and depth in different kinds of competition.

To perform the fit we consider again a data set represented by its adjacency matrix $\boldsymbol{A}$ and write the data likelihood in the form of Eq. (4.3):

$$P(\boldsymbol{A}|\boldsymbol{s},\alpha,\beta) = \prod_{ij} f_{\alpha\beta}(s_i - s_j)^{A_{ij}}. \tag{4.14}$$

The scores $\boldsymbol{s}$ are assumed to have the Gaussian prior of Eq. (4.5), and we assume a uniform (least informative) prior on α, which means $P(\alpha) = 1$. We cannot use a uniform prior on β, since it has infinite support, so instead we use a prior that is approximately uniform over "reasonable" values of β and decays in some slow but integrable manner outside this range. A suitable choice in the present case is the half-Cauchy distribution

$$P(\beta) = \frac{2w/\pi}{\beta^2 + w^2}, \tag{4.15}$$

where w controls the scale on which the function decays. In this chapter we use $w = 4$, which roughly corresponds to the range of variation in β that we see in real-world data sets, and has the convenient property of giving a uniform prior on the angle of $f_\beta(s)$ at the origin.

It is worth mentioning that the choice of prior on β *does* have an effect on the results in some cases. When data sets are large and dense, priors tend to have relatively little impact because the posterior distribution is narrowly peaked around the same set of values no matter what choice we make. But some of the data sets we study here are quite sparse and for these the results can vary with the choice of prior. Our qualitative conclusions remain the same in all cases, but it is worth bearing in mind that the quantitative details can change.

Combining the likelihood and priors, we now have

$$P(\boldsymbol{s}, \alpha, \beta | \boldsymbol{A}) = P(\boldsymbol{A} | \boldsymbol{s}, \alpha, \beta) \frac{P(\alpha)P(\beta)P(\boldsymbol{s})}{P(\boldsymbol{A})}. \tag{4.16}$$

The prior on $\boldsymbol{A}$ is unknown but constant, so it can be ignored. We now draw from the distribution $P(\boldsymbol{s}, \alpha, \beta | \boldsymbol{A})$ to obtain a representative sample of values $\boldsymbol{s}, \alpha, \beta$. In our calculations we generate the samples using the Hamiltonian Monte Carlo method [94] as implemented in the probabilistic programming language Stan [15], which is ideal for sampling from continuous parameter spaces such as this. The running time to obtain the samples depends on the computational cost per iteration, which is proportional to the number of matches m, and on the Monte Carlo mixing time, which is roughly proportional to the number of competitors n. The total running thus scales roughly as $\mathrm{O}(mn)$. In practice, a few thousand samples are sufficient to get a good picture of the distribution of α and β, which in our implementation takes anywhere from a few seconds to an hour or so for our largest data sets.

4.3.4 Minimum violations ranking

One special case of our model worth mentioning is the limit $\beta \to \infty$ for fixed $\alpha > 0$. In this limit the function $f_{\alpha\beta}(s)$ becomes a step function with value

$$f_{\alpha,\infty}(s) = \begin{cases} \frac{1}{2}\alpha & \text{if } s < 0, \\ \frac{1}{2} & \text{if } s = 0, \\ 1 - \frac{1}{2}\alpha & \text{if } s > 0. \end{cases} \tag{4.17}$$

For this choice the data likelihood becomes

$$P(\boldsymbol{A}|\boldsymbol{s},\alpha,\beta) = \left(\tfrac{1}{2}\alpha\right)^{v}\left(1-\tfrac{1}{2}\alpha\right)^{m-v}, \tag{4.18}$$

where m is the total number of games/interactions/comparisons and v is the number of "violations," meaning games where the weaker player won. Then the log-likelihood is

$$\begin{aligned}\log P(\boldsymbol{A}|\boldsymbol{s},\alpha,\beta) &= -v\log\frac{1-\frac{1}{2}\alpha}{\frac{1}{2}\alpha} + m\log\left(1-\tfrac{1}{2}\alpha\right) \\ &= -Av - B, \end{aligned}\tag{4.19}$$

where A and B are positive constants. This log-likelihood is maximized when the number of violations v is minimized, recovering the *minimum violations ranking* of Section 1.2.4, the ranking such that the minimum number of games are won by the weaker player. Thus the minimum violations ranking can be thought of as the limit of our model in the special case where $\beta \to \infty$.

Just as we had defined models in Chapter 2 that generalize many SBMs for identifying group structure, our model here includes both the standard Bradley-Terry model and minimum violation ranking as special cases, allowing us to directly compare them on real data sets.

4.4 Results

We have applied these methods to a range of data sets representing competition in sports and games as well as social hierarchies in both humans and animals. The data sets we study are listed in Table 4.1.

Figure 4.2 summarizes our results for the posterior probability density of the luck and depth parameters. The axes of the figure indicate the values of α and β and each cloud is an estimate of $P(\alpha,\beta|\boldsymbol{A})$, computed as a kernel density estimate from Monte Carlo sampled values of α and β. The + signs in the figure represent the mean values of α and β for each data set computed directly by averaging the samples.

The figure reveals some interesting trends. Note first that all of the sports and games—chess, basketball, video games, etc.—appear on the left-hand side of the plot in the region of low depth of competition, while all the social hierarchies are on the right with higher depth. We conjecture that the low depth of the sports and games is a result of a preference for matches to be between roughly evenly matched opponents, as discussed

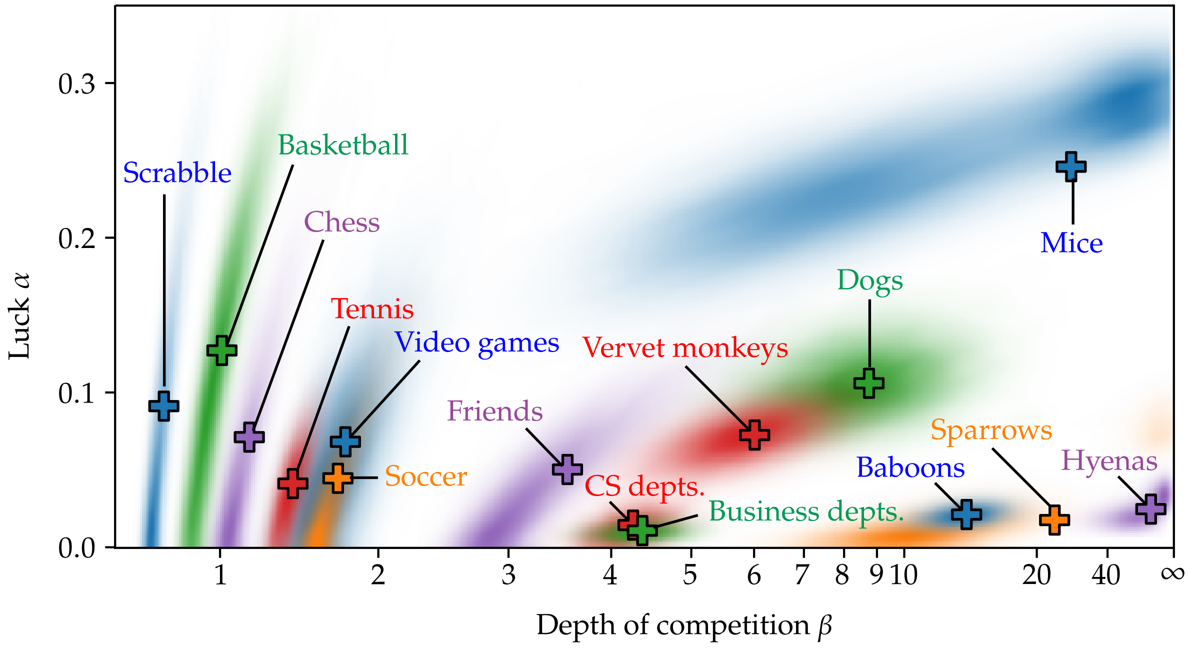


Figure 4.2: Posterior distributions of luck and depth. Each cloud represents the posterior distribution $P(\alpha, \beta|\boldsymbol{A})$ of the luck and depth parameters for a single data set, calculated from the Monte Carlo sampled values of α and β using a Gaussian kernel density estimate. The + signs indicate the expected values $\hat{\alpha}, \hat{\beta}$ of the parameters for each data set.

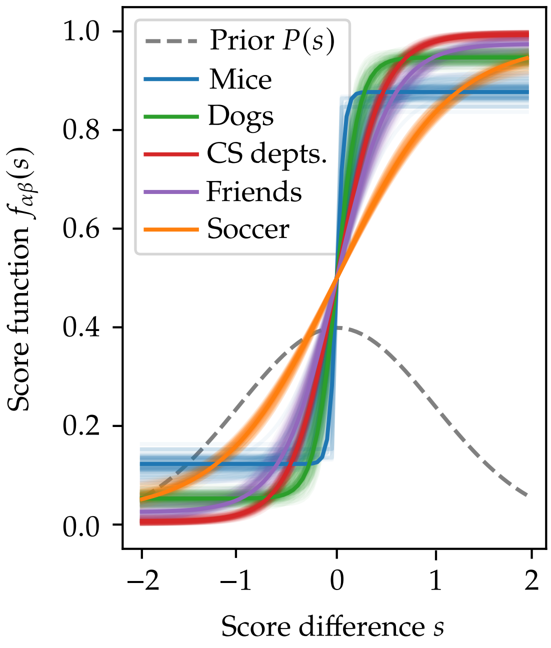


Figure 4.3: Fitted functions $f_{\alpha\beta}(s)$ for a selection of the data sets. The bold curve in each case corresponds to the expected values $\hat{\alpha}, \hat{\beta}$, while the other surrounding curves are for a selection of values sampled from the posterior distribution, to give an idea of the variation around the average.

	Data set	$\hat{\beta}$	n	m	Description	Ref.
Sports/games	Scrabble	0.68	587	23477	*Scrabble* tournament matches 2004–2008	[4]
	Basketball	1.01	240	10002	National Basketball Association games 2015–2022	[79]
	Chess	1.17	917	7007	Online chess games on lichess.com in 2016	[3]
	Tennis	1.44	1272	29397	Association of Tennis Professionals men's matches 2010–2019	[119]
	Soccer	1.73	1976	7208	Men's international association football matches 2010–2019	[68]
	Video games	1.77	125	1951	*Super Smash Bros Melee* tournament matches in 2022	[5]
Human	Friends	3.54	774	2799	High-school friend nominations	[128]
	CS depts.	4.25	205	4388	PhD graduates of one department hired as faculty in another	[31]
	Business depts.	4.36	112	7856	PhD graduates of one department hired as faculty in another	[31]
Animal	Vervet monkeys	6.01	41	2930	Dominance interactions among wild vervet monkeys	[130]
	Dogs	8.74	27	1143	Aggressive behaviors in a group of domestic dogs	[121]
	Baboons	13.19	53	4464	Dominance interactions among a group of captive baboons	[55]
	Sparrows	22.92	26	1238	Attacks and avoidances among sparrows in captivity	[133]
	Mice	26.48	30	1230	Dominance interactions among mice in captivity	[138]
	Hyenas	100.58	29	1913	Dominance interactions among hyenas in captivity	[124]

Table 4.1: Data sets in order of increasing depth of competition β. Here n is the number of participants and m is the number of matches/interactions. Further information on the data sets is given in Appendix D.19.

in Section 4.3.2. For a game to be entertaining to play or watch the outcome of matches should not be too predictable, but in a sport or league with high depth the average pairing is very uneven, with the stronger player very likely to win. Low depth of competition ensures that matches are unpredictable and hence entertaining. In games such as chess, which have high intrinsic depth, the depth can be reduced by restricting tournaments to players in a narrow skill range, such as world-class players, and this is commonly done in many sports and games. We explore this interpretation further in Appendix D.22.

There are no such considerations at play in social hierarchies. Such hierarchies are not, by and large, spectator sports, and there is nothing to stop them having high depth of competition. The results in Fig. 4.2 indicate that in general they do, though the animal hierarchies are deeper than the human ones. A high depth in this context indicates a hierarchy in which the order of dominance between the typical pair of competitors is clear. This accords with the conventional wisdom concerning hierarchies of both humans and animals, where it appears that participants are in general clear about the rank ordering.

Another distinction that emerges from Fig. 4.2 is that the results for sports and games generally do not give strong support to a nonzero luck parameter. The expected values, indicated by the + signs, are nonzero in most cases, but the clouds representing the posterior distributions give significant weight to points close to the $\alpha = 0$ line, indicating that we cannot rule out the possibility that $\alpha = 0$ in these competitions. For many of the social hierarchies, on the other hand, there is strong evidence for a nonzero amount of luck, with the posterior distribution having most of its weight well away from $\alpha = 0$, a finding that accords with our intuition about social hierarchies. There would be little point in having any competition at all within a social hierarchy if the outcomes of all contests were foregone. If participants knew that every competitive interaction was going to end with the higher-ranked individual winning and the lower-ranked one backing down, then there would be no reason to compete. It is only because there is a significant chance of a win that competition occurs at all.

An interesting counter-example to this observation comes from the two faculty hierarchies, which represent hiring practices at US universities and colleges. The interactions in this data set indicate when one university hires a faculty candidate who received their doctoral training at another university, which is considered a win for the university where the candidate trained. The high depth of competition and low luck parameter for these data sets indicates that there is a pronounced hierarchy of hiring with a clear pecking order and that the pecking order is rarely violated. Lower-ranked universities hire the graduates of higher-ranked ones, but the reverse rarely happens.

Figure 4.3 shows a selection of the fitted functions $f_{\alpha\beta}(s)$ for five of the data sets. For each data set we show in bold the curve for the expected values $\hat{\alpha}, \hat{\beta}$ along with ten other curves for values of α, β sampled from the posterior distribution, to give an indication of the amount of variation around the average. We see for example that the curve for the soccer data set has a shallow slope (low depth of competition) but is close to zero and one at the limits (low luck). The curve for the mice data set, by contrast, is steep (high depth) but clearly has limits well away from zero and one (nonzero luck).

4.4.1 Luck and parameter identifiability

Inherent in the view of competition that underlies our model are two different types of randomness. There is the randomness inherent in the probabilistic nature of the model: even when one player is better than the other there is always a chance they may lose, so long as the players' levels of skill are not too severely imbalanced. But there is also the randomness introduced by the luck parameter, which applies no matter how imbalanced the players are, even if one is infinitely better than the other.

In a low-depth situation it can be difficult to distinguish between these two types of randomness. When depth is low there are few (or no) players who are very good or very bad, so there are few matches were a good player is unequivocally observed to lose because of the element of "luck." In mathematical terms, the score function $f(s)$ in a low-depth competition is shallow in its central portion, close to the origin, and moreover it is only this portion that gets probed by the matches, since there are few contests between badly mismatched players. But a score function with a shallow center can be generated either by a large value of α or a small value of β—the functional forms are very similar either way.

In practice this means that the values of α and β suffer from poor identifiability in this low-depth regime. This is visible in Fig. 4.2 as the long, thin probability clouds of the sports and games on the left-hand side of the plot. For these there is a set of parameter value pairs α, β that fall roughly along a curve in the plot and that all have similar posterior probability, and hence it becomes difficult to pin down the true parameter values. This phenomenon particularly affects the luck parameter α, whose spread is so broad in this regime that we cannot reliably determine whether it is nonzero.

As depth increases, on the other hand, we expect that there will be a larger number of competitors who are either very strong or very weak, and from the outcomes of their matches we can determine the level of luck with more certainty. This is reflected in the distributions on the right of Fig. 4.2, for many of which it is possible to say clearly that α is nonzero.

An alternative view of the same behavior is that the long thin probability clouds in the figure imply the existence of a particular combination of luck and depth that is narrowly constrained for each data set, and an orthogonal combination that is highly uncertain. In Appendix D.23, we define a measure of "predictability" of competition in terms of the amount of information needed to communicate the outcomes of all matches in a data set, and show that this predictability corresponds precisely to the narrowly defined direction in the figure, so that predictability can be estimated accurately in all cases, even when there is considerable uncertainty about the raw parameters α and β.

4.5 Predicting wins and losses

In addition to allowing us to infer the luck and depth parameters and rank competitors, our model can also be used to predict the outcomes of unobserved matches. If we fit the model to data from a group of competitors, we can use the fitted model to predict the winner of a new contest between two of those same competitors. The ability to accurately perform such predictions can form the basis for consumer product recommendations and marketing, algorithms for guiding competitive strategies in sports and games, and the setting of odds for betting, among other things.

We can test the performance of our model in this prediction task using a cross-validation approach. For any data set $\boldsymbol{A}$ we randomly remove or "hold out" a small portion of the matches or interactions and then fit the model to the remaining "training" data set. Then we use the fitted model to predict the outcome of the held-out matches and compare the results with the actual outcomes of those same matches.

The simplest version of this calculation involves fitting our model to the training data by making point estimates of the parameters and scores. We first estimate the expected posterior values $\hat{\alpha}, \hat{\beta}$ of the parameters given the training data. Then, given these parameter values, we maximize the posterior probability as a function of $\boldsymbol{s}$ to obtain MAP estimates $\hat{\boldsymbol{s}}$ of the scores. Finally, we use the combined parameter values and scores to calculate the probability $\hat{p}_{ij} = f_{\hat{\alpha}\hat{\beta}}(\hat{s}_i - \hat{s}_j)$ that a held-out match between i and j was won by i, with $f_{\alpha\beta}(s)$ as in Eq. (4.13). Further discussion of the procedure is given in Appendix D.21.

We can quantify the performance of our predictions by computing the log-likelihood of the actual outcomes of the held-out matches under the predicted probabilities $\hat{p}_{ij}$. If W_{ij} is the number of times that i actually won against j then the log-likelihood per game is

$$Q = \frac{\sum_{ij} W_{ij} \log \hat{p}_{ij}}{\sum_{ij} W_{ij}}. \tag{4.20}$$

This measure naturally rewards cases where the model is confident in the correct answer ($\hat{p}_{ij}$ close to 1) and heavily penalizes cases where the model is confident in the wrong answer ($\hat{p}_{ij}$ close to 0). Note that the log-likelihood is equal to minus the description length of the data—the amount of information needed to describe the true sequence of wins and losses in the held-out data given the estimated probabilities $\hat{p}_{ij}$—so models with high log-likelihood are more parsimonious in describing the true pattern of wins and losses. (An alternative way to quantify performance would be simply to compute the fraction of correct predictions made by each model. Some results from this approach are given in Appendix D.20, and are largely in agreement with the results for log-likelihood.)

To place the performance of our proposed model in context, we compare it against a basket of other ranking models and methods, including widely used standards, some recently proposed approaches, and some variants of the approach proposed in this chapter. As a baseline we compare performance against the standard Bradley-Terry model with a logistic prior, which is commonly used in many ranking tasks, particularly in sports, and which we have ourselves used and recommended in the past [100]. We measure the performance of all other models against this one by calculating the difference in the log-likelihood per match, Eq. (4.20). The other models we test are:

1. The luck-plus-depth model of this chapter.

2. A depth-only variant in which the parameter α is set to zero.

Model	α	β
Luck-plus-depth	inferred	inferred
Depth only	0	inferred
Luck only (min. violations)	inferred	∞
B-T, max. likelihood	0	100
B-T, logistic prior	0	≈ 2.56

Table 4.2: Summary of the parameter values of the joint luck-plus-depth model presented in this paper that correspond to the other models it generalizes. We note that here the Bradley-Terry (B-T) model with the logistic prior is only approximately represented by our model with $\beta \approx 2.56$, as detailed in Appendix D.22.

3. A luck-only variant in which the parameter β is set to ∞, which is equivalent to minimum violations ranking.

4. The Bradley-Terry model under maximum-likelihood estimation, which is equivalent to imposing an improper uniform prior. Note that maximum-likelihood estimates diverge if a player wins (or loses) all of their matches and to avoid this, in keeping with previous work [40], we impose a very weak L2 regularization of the scores which is equivalent to a MAP estimate with Gaussian prior of width $\sigma = 100$.

5. The "SpringRank" model of De Bacco *et al.* [40], which ranks competitors using a physically motivated mass-and-spring model.

Apart from the SpringRank model, all of these alternative model are (or at least approximately are) contained as special cases of the full luck-plus-depth model. Table 4.2 summarizes the parameter values that correspond to these models. This property also then allows us to compare the Bayesian evidence of these models using the form of the posterior distribution of parameters in Figure 4.2. As discussed earlier, depending on the data set different models are excluded based upon whether they meaningfully intersect with the relevant space of parameters.

The models we can construct and compare in this way are meant to be a representative selection of ranking models but not comprehensive, excluding for instance models that incorporate information beyond wins and losses, and multidimensional models [28, 85]. In these predictive cross-validation tests, the proportion of data held out was 20% in all cases, chosen uniformly at random, and at least 50 random repetitions of the complete process were performed for each model for each of the data sets listed in Table 4.1.

The results are summarized in Fig. 4.4. The horizontal dashed line in the figure represents the baseline set by the Bradley-Terry model and the points with error bars represent the increase (or decrease) in log-likelihood relative to this level for each model and data set. The error bars represent the upper and lower quartiles of variation of the results over the random repetitions. (We use quartiles rather than standard deviations because the distributions are highly non-normal in some cases.)

We note a number of things about these results. First, the model of this chapter performs best on our tests for every data set without exception, within the statistical uncertainty, although the depth-only version of the model is also competitive in many cases, particularly for the sports and games. The latter observation is unsurprising, since, as we have said, there is little evidence for $\alpha > 0$ in the games. For the particular case of the dominance hierarchy of hyenas, the minimum violations ranking is competitive, which is also unsurprising: as shown in Fig. 4.2 this hierarchy is very deep—the value of β is over 100—and hence there is little difference between our model and the minimum violations ranking. For all the other data sets the minimum violations ranking performs worse—usually much worse—than our model. (Arrows at the bottom of the figure indicate results so poor they fall off the bottom of the scale.) The maximum likelihood fit to the Bradley-Terry model also performs quite poorly, a notable observation given that this is one of the most popular ranking algorithms in many settings. It even performs markedly worse than the same Bradley-Terry model with a logistic prior. Finally, we note that the SpringRank algorithm of [40] is relatively competitive in these tests, though it still falls short of the model of this chapter and the standard Bradley-Terry model with logistic prior.

As mentioned above, our selection of models excludes multidimensional models, which have substantially larger parameter spaces and allow for a wider range of behaviors, such as intransitive competition, and which could in principle provide better fits to the data. In other tests (not shown here) we found one such model, the blade-chest model of Chen and Joachims [28], that outperforms our model on four of the animal data sets (dogs, baboons, sparrows, and hyenas), although it performs poorly in most other cases. This could suggest the presence of intransitivity in these data sets.

4.6 Conclusion

In this chapter we have studied the ranking of competitors based on pairwise comparisons between them, as happens for instance in sports, games, and social hierarchies. Building on the standard Bradley-Terry ranking model, we have extended the model to include two additional features: an element of luck that allows weak competitors to occasionally beat strong ones, and a "depth of competition" parameter that captures the number of distinguishable levels of play in a hierarchy. Deep hierarchies with many levels correspond to complex games or social structures. We have fitted the proposed model to data sets representing social hierarchies among both humans and animals and a range of sports and games, including chess, basketball, soccer, and video games. The fits give us estimates of the luck and depth of competition in each of these examples and we find a clear pattern in the results: sports and games tend to have shallow depth and little evidence of a luck component, while social hierarchies are significantly deeper and more often have an element of luck, with the animal hierarchies being deeper than the human ones.

We also tested our model's ability to predict the outcome of contests. Using a cross-validation approach we found that the model performs as well as or better than every other model tested in predictive tasks and very significantly better than the most common previous methods such as maximum likelihood fits to the Bradley-Terry model or minimum violations rankings.

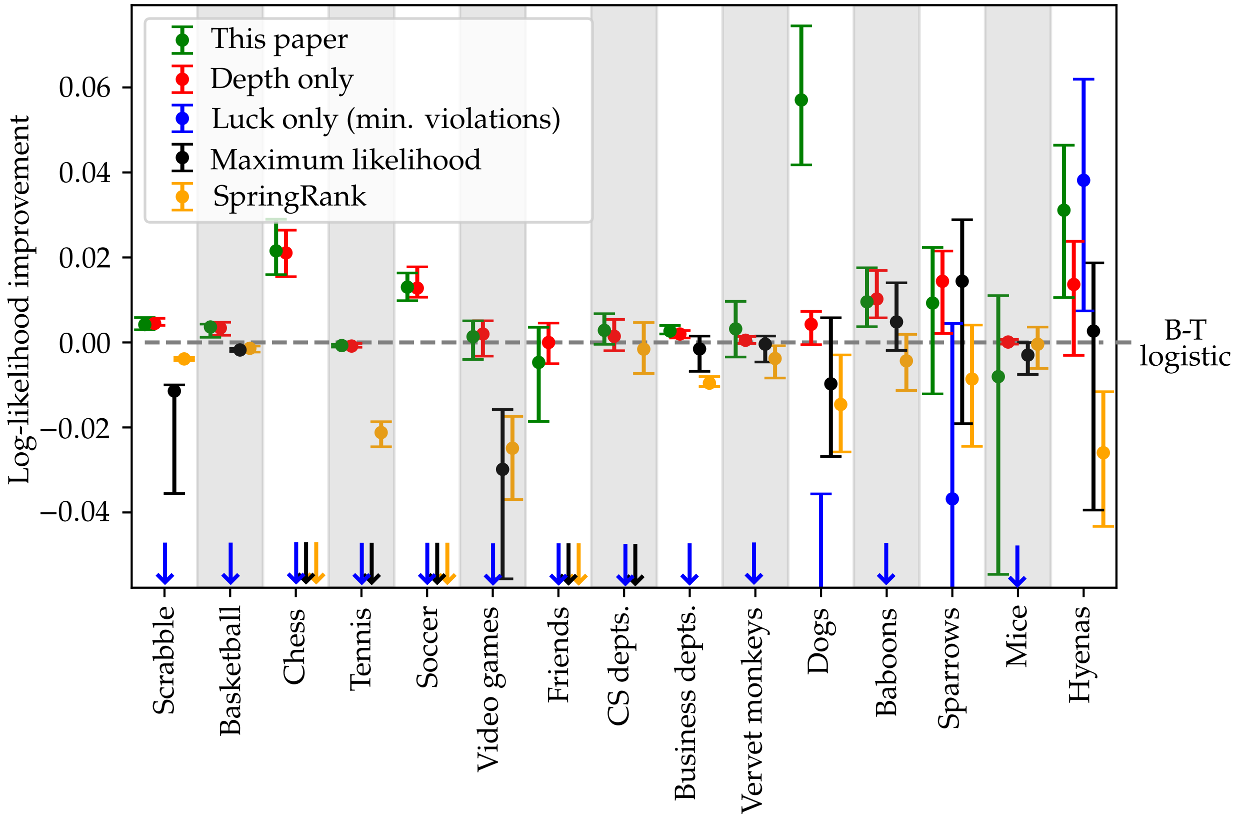


Figure 4.4: Comparative performance of the model of this chapter and a selection of competing models and methods, in the task of predicting the outcome of unobserved matches in a cross-validation experiment. Performance is measured in terms of the log-likelihood (base 2) of the actual outcomes of matches within the fitted model, which is also equal to minus the description length in bits required to transmit the win/loss data given the fitted model. Log-likelihoods are plotted relative to that of the standard Bradley-Terry model with a logistic prior (the horizontal dashed line). Error bars represent upper and lower quartiles over at least 50 random repetitions of the cross-validation procedure in each case. The arrows along the bottom of the plot indicate cases where the log-likelihood is outside the range of the plot.

Chapter 5

Conclusion

Throughout this thesis, we have leveraged tools from diverse disciplines to model networks and enhance our understanding of the complex systems they represent. We observed rich structures of groups and hierarchies and developed methods to measure their underlying mechanisms. Our work not only provided explanations for observed patterns but also sought to justify and validate these inferences.

In the Introduction we reviewed examples of the systems we study and noted how these sources' properties leave imprints of group and hierarchy structure on networks of interactions, motivating us to create tools to infer these processes. We described how insights from statistics, physics, computer science, and machine learning, reviewed in Appendix A, inform network models to elucidate these structures. Notably, we discussed the concept of building nested models, integrating multiple plausible models for direct Bayesian comparisons. This led us to the *general configuration model*, which facilitates comparison between Erdős-Rényi and configuration models and measures degree distribution inequality. We also examined stochastic block models (SBM) and Bradley-Terry (BT) models, fundamental for understanding group and hierarchy structures.

Chapter 2 focused on extending the stochastic block model to better reflect group structures in networks. The first extension, the *general degree-corrected SBM*, accommodates a broader range of degrees within groups than the usual SBM by substituting its Erdős-Rényi base with the general configuration model, both improving predictions and enabling the measurement of intra-group degree inequality. The second refinement introduced the *simple* and *general* assortative SBMs, concentrating on nature of the group structure rather than the degree distribution. By explicitly modeling assortativity, these models overcome the resolution limit that prevents other methods from identifying small groups. The general assortative SBM, further captures variations in group connections, revealing that real network groups are more idiosyncratic than previously assumed, and so outperforms traditional models by up to 20% in predictive tests.

Chapter 3 focused on analyzing the outputs of these algorithms, comparing identified group structures to known groups or other contextual clusterings. We adopted an information-theoretic framework to compute the mutual information shared between two labelings of the same set of objects. We identified two major shortcomings in the typical mutual information. First, that it tends to favor labelings with too many groups because it neglects a contingency table term in the transmission process. We introduced a more accurate approximation of this cost and, consequently, a more precise measure of mutual information. In Section 3.2, we demonstrated the bias introduced by the typical symmetric normalization of mutual information and proposed an asymmetric normalization to address it. We demonstrated the improved measure's utility by evaluating various community detection methods on synthetic networks.

In Chapter 4, we shifted our focus to analyzing hierarchies in directed networks. By expanding on the traditional Bradley-Terry model, we quantified both the luck inherent in a hierarchy—where even the least capable participant might beat the best—and the hierarchy's depth, the overall number of levels of play or strictness of a ranking. Applying this model to real datasets, we observed significant variations across different contexts. Sports and games display competitive low-depth hierarchies, animal social hierarchies show high strictness, and human hierarchies generally fall in between. We also found that only animal hierarchies have have an appreciable element of luck. Utilizing this richer model, we achieved a more detailed understanding of these systems, enhancing both predictive accuracy and parsimonious modeling.

Although we have advanced the simple models we began the thesis with into a more comprehensive and descriptive picture, there are still clear areas for improvement. For one, our models do not capture the local structure of networks, for example triadic closure: the tendency for two friends of a person to also be friends with each other. Such local behaviors can profoundly influence the system, affecting processes like disease transmission along edges. Recent tools like hyperbolic random graphs [74] and loopy belief propagation [22] incorporate these behaviors but mainly focus on getting this local structure correct. A promising direction for future work could be to develop models that integrate non-trivial local structures with the global properties targeted in this thesis. Additionally, the models we have described focus on relatively simple networks, lacking dynamic information or additional contexts provided by weighted graphs or hypergraphs. Extending this thesis's techniques to these more decorated networks could enhance understanding of how context and structure interrelate.

Overall, this thesis has demonstrated the broad applicability of network science and advanced two crucial tools in the field. By sharpening these techniques we hope to facilitate future endeavors to understand complex systems.

Appendix A

Supplementary Background Material

In this thesis we invoke tools across many disciplines to probe patterns of groups and hierarchies in network data. While these methods are particularly useful to network science, they apply to a far more general context. As such in this appendix we review these subjects by broadening our focus to consider generic data sets and toy models. Over Appendices A.1 to A.4 we consider the simple example of flipping a coin from the perspectives of statistics, physics, computer science, and machine learning in turn. Figure A.1 contains cartoons of these varied perspectives on the simple system. Analogously, Figure A.2 illustrates common examples of where these same concepts appear within network science and this thesis.

A.1 Statistical inference

Suppose we perform a simple experiment and record whether a two-sided coin lands "heads" or "tails" over n flips. Before the experiment begins we might hypothesize that we hold a fair coin: that all coin flips are independent and result in heads with probability $p = 0.5$ and tails with probability $1 - p = 0.5$. This natural assumption defines a *model* by assigning a *likelihood* of observing any particular sequence of heads and tails over the n trials

$$P(\overbrace{\mathrm{H}, \ldots, \mathrm{T}}^{n \text{ flips}} | p = 0.5) = \frac{1}{2^n}. \tag{A.1}$$

Although all possible sequences of n outcomes are equally likely to appear under this model, certain observations may lead us to doubt our initial hypothesis. Suppose every coin flip we observe lands on heads. Although a single heads does not raise suspicion, 10 heads in a row start to be quite surprising at a probability of $\sim 0.1\%$. And after 100 fair heads in a row we may start to question our place in the universe. The question arises:

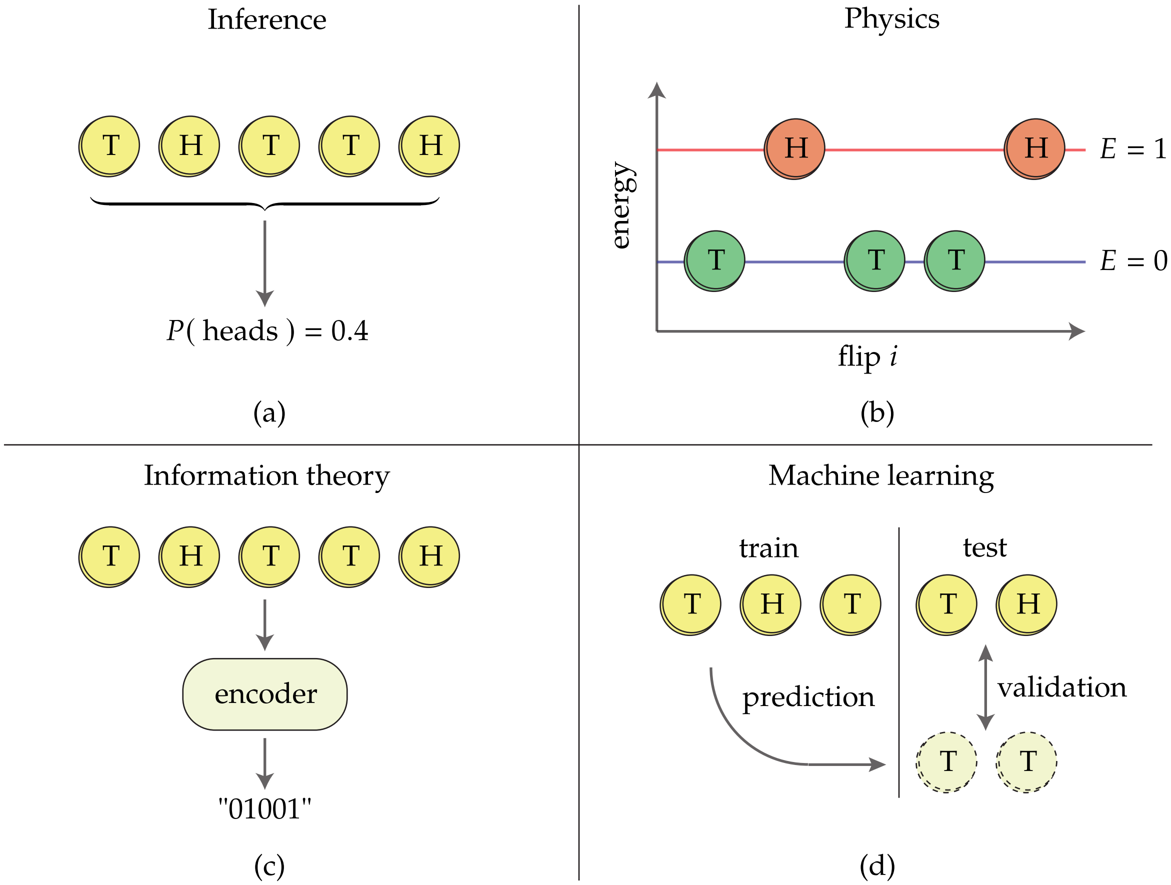


Figure A.1: Schematics of four perspectives on coin flips. (a) Using statistics to infer the probability of landing heads ("H"). (b) Physical interpretation where tails ("T") is more likely as a lower "energy" state than heads. (c) Encoding the sequence of flips as a binary string to quantify the complexity of the sequence in bits. (d) Cross-validation of a model trained on a subset of the data against a withheld testing set.

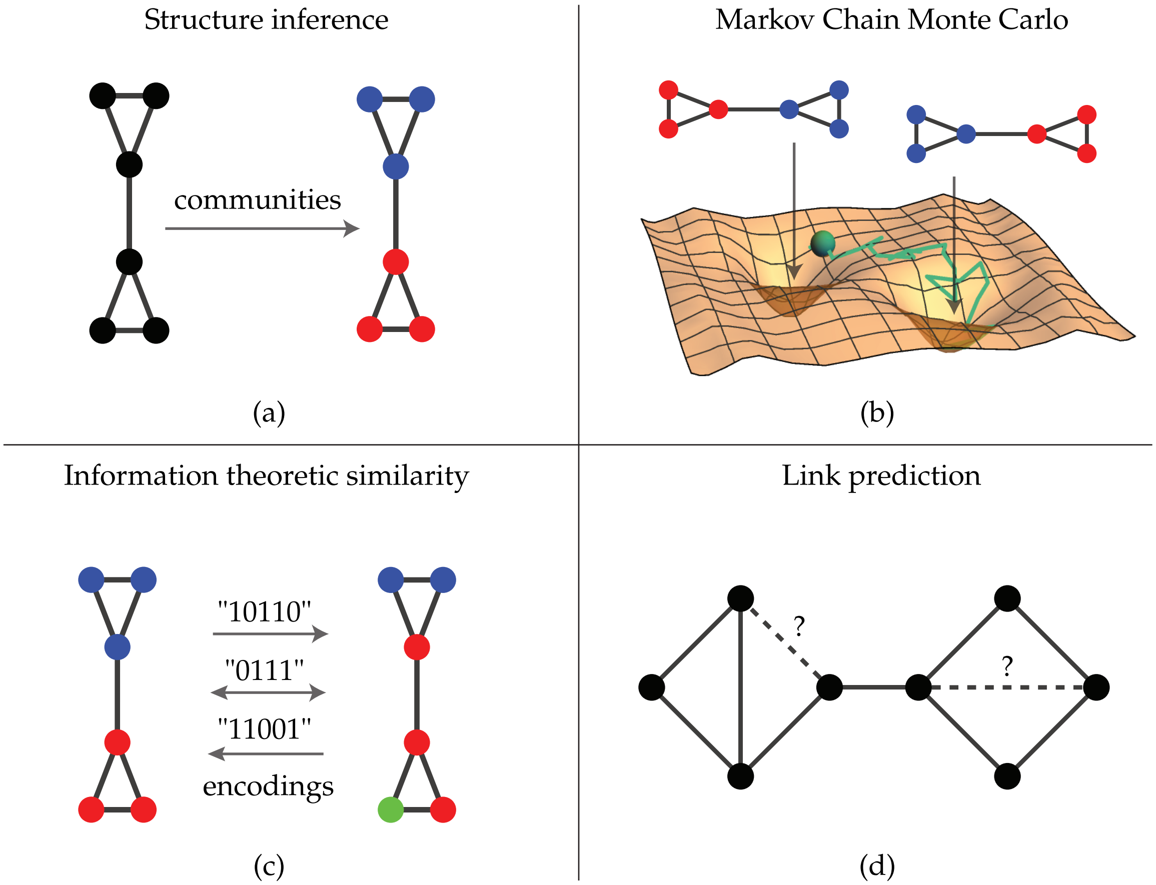


Figure A.2: Example applications of the tools discussed in this section to network science. (a) Statistical inference of community structure of a given network, as in Section 1.2.3. (b) Markov Chain Monte Carlo to explore the posterior distribution (energy landscape) of possible community structures of a network, as in Appendix B.6. (c) Information theoretic measures of the information content and similarity of two possible network community structures, as in Chapter 3. (d) Prediction of unobserved links in a network, as in Figure 2.4.

how many heads should we observe before abandoning our initial conjecture? And if the coin is not fair, what is its nature? Statistics provides many frameworks for approaching these questions.

In the *frequentist* approach our initial assumption of fairness is called a *null model* (or "null hypothesis") which our experiment may then "reject." In this picture we define a *test statistic* that captures some surprising aspect of our observed data, in this case the unusually large number of heads $n_H = n$ observed. We then compute the likelihood that the null model could produce a result with a test statistic at least as extreme as that observed, known as the *p-value*. If this p-value is small it is unlikely the null model alone could generate an observation similar to ours. This improbability suggests our model is lacking and serves as grounds to "reject" it in favor of some alternative model more likely to have produced our observation.

In linear regression, these p-values are often reported to reject the possibility of a slope of 0, of no relation between two variables. In network science, simple null models are likewise used to demonstrate that some observed structural feature like community or hierarchy requires a richer model to reproduce, such as those discussed in Section 1.2.

For a fair coin the probability of observing n_H heads out of n trials is given by the binomial distribution

$$P(n_H | p = 0.5) = \frac{1}{2^n} \binom{n}{n_H}. \tag{A.2}$$

The p-value, the chance of observing some number of heads n'_H greater than or equal to n_H, is then

$$P(n'_H \geq n_H | p = 0.5) = \frac{1}{2^n} \sum_{n'_H \geq n_H} \binom{n}{n'_H}. \tag{A.3}$$

As shown in Figure A.3, over 10 fair coin flips we expect an average of 5 heads in this model although small fluctuations such as 4 or 6 heads are also common. More lopsided outcomes are increasingly unlikely, so that the p-values of observing larger n_H approach zero.

A smaller p-value yields a more *statistically significant* rejection of the null hypothesis. A threshold of $P < 0.05$ is often used as a minimum standard, in which case we would reject the hypothesis of a fair coin in our experiment if 9 or 10 of the coin flips land heads. This demarcation at 0.05 is arbitrary; however small there is always some chance that a genuine fair coin produced an observation as unusual as the one made. We would expect to obtain a p-value less than 0.05 and so reject the model in 1 of 20 experiments conducted

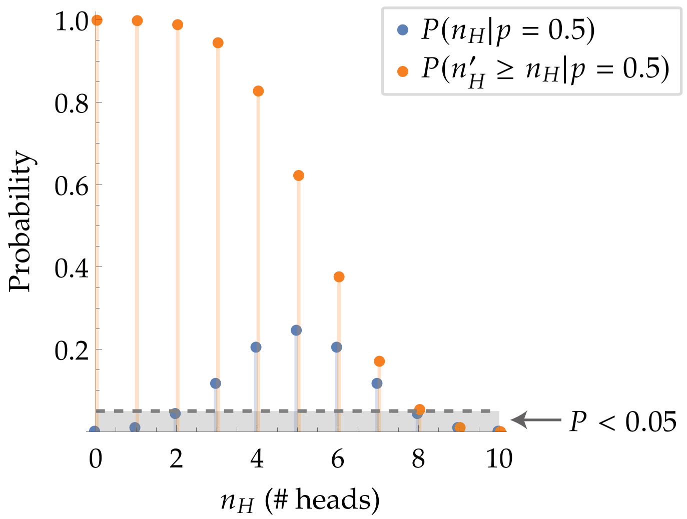


Figure A.3: Distribution of the number of heads n_H observed over $n = 10$ flips of a fair coin and the resulting p-value probabilities of observing at least n_H heads. The region where the p-value is less than 0.05 is plotted in gray, a threshold to reject the null hypothesis of a fair coin.

with even a truly fair coin. The choice of p-value threshold reflects a tolerance for this possibility that our rejection is the result of chance rather than a genuine signal.

If such significance testing has led us to doubt that the coin is fair, we may consider an alternative hypothesis. We could instead model a biased coin whose flips are still independent but land heads with some fixed probability $p \in [0, 1]$ and tails with probability $1 - p$. Under this assumption the likelihood of a sequence with n_H heads and $n_\mathrm{T} = n - n_\mathrm{H}$ tails is

$$P(\overbrace{\mathrm{H}, \ldots, \mathrm{T}}^{n_\mathrm{H}\text{ heads}} | p) = p^{n_\mathrm{H}}(1 - p)^{n - n_\mathrm{H}}. \tag{A.4}$$

This defines a *nested model* by generalizing the fair coin as the special case $p = 0.5$. Thus the original fair coin model can be directly compared against other choices of the *parameter p* which each represent possible coins of varying levels of bias towards landing heads or tails. Before the experiment begins we imagine all these coins are possible and aim to *infer* the "true" value of p realized by our coin based on the observations.

To apply this model suppose that after 10 trials we observe a sequence

$$(\mathrm{H}, \mathrm{H}, \mathrm{T}, \mathrm{T}, \mathrm{H}, \mathrm{T}, \mathrm{T}, \mathrm{H}, \mathrm{T}, \mathrm{T}), \tag{A.5}$$

now a mixture of $n_\mathrm{H} = 4$ heads and $n_\mathrm{T} = 6$ tails. We fit the model likelihood Eq. (A.4) to this data by finding the value of p which would maximize the probability that our observed sequence occurred. For this model this is simply equal to the observed fraction of heads

$$\hat{p}_{\mathrm{ML}} = \frac{n_{\mathrm{H}}}{n} \tag{A.6}$$

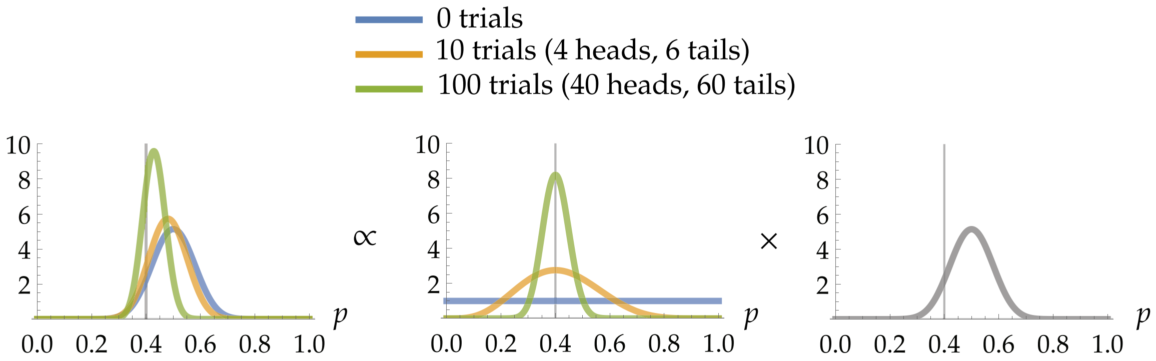
where the "ML" subscript indicates that this is the *maximum likelihood* estimate of p. Our sequence of observations (A.5) gives the estimate $\hat{p}_{\mathrm{ML}} = 0.4$.

Although this $\hat{p}_{\mathrm{ML}}$ is the single parameter value most likely to have generated our observations, it is unclear how seriously we should take the estimate. If the coin was in fact fair, $p = 0.5$, the slight imbalance towards tails we observe could very well be a random fluctuation in our small sample size as seen in Figure A.3. In our earlier language the p-value is not small enough to warrant rejecting the null hypothesis of a fair coin. To conclude that the one "true" value of p is 0.4, for example to predict that 400 of the next 1000 flips will be heads, would likely *overfit* the data. As a more extreme example if we only observe a single coin flip which happens to land heads, the maximum likelihood estimate would conclude that $\hat{p}_{\mathrm{ML}} = 1$ and thus that the coin will always land heads. There should be uncertainty in our conclusions, particularly when they are based on such little evidence.

By adopting a *Bayesian* perspective we can naturally represent this uncertainty in our inferences. In this framework we critically never exclude the possibility of any potential parameter value. Rather we infer from the data that some parameter values are more probable than others. At each stage of the experiment we represent our *belief* of likely values as a distribution over all possibilities rather than a single best-fit point.

Before considering the data we define a *prior distribution* (or "prior") over the parameters that reflects our initial assumptions. Depending on the context of our experiment we may have quite different expectations that will rightfully influence the conclusions we draw. For example if we use a standard legal tender coin we may hesitate to conclude that the coin is notably biased, even after observing 10 heads in a row. However, if the flips instead involve either a hypothetical or peculiar-looking coin there may be more room for doubt. The form of the prior distribution precisely specifies our initial assumptions.

For our example we will adopt a prior assumption that the coin is likely to be fair or approximately fair. Although a physical coin could conceivably be as biased as $p = 0.4$, we expect a balanced $p = 0.5$ to be far more likely. To represent this we define a prior



(a) posterior $P(p|\mathrm{H}, ..., \mathrm{T})$ (b) likelihood $P(\mathrm{H}, ..., \mathrm{T}|p)$ (c) prior $P(p)$

Figure A.4: (a) Posterior distributions of the parameter p after observing 0 coin flips (a.k.a. prior distribution), 10, and 100 coin flips. In each experiment 40% of trials are heads, a ratio marked by the vertical line in each plot. The posterior distributions are proportional to the product of the likelihood (b) and prior (c) over p. As more observations are made the posterior is increasingly informed by the model likelihood.

distribution over possible p peaked at $p = 0.5$, particularly a beta distribution $p \sim B(20, 20)$

$$P(p) = \frac{41!}{20!20!} p^{20}(1 - p)^{20}, \tag{A.7}$$

plotted in Figure A.4c. This particular form of the prior and the number 20 are arbitrarily chosen for this example, although they represent a reasonable assumption for this scenario.

As we observe coin flips we update these prior beliefs to reflect new data and arrive at a new *posterior* distribution of likely parameter values. Using Bayes' law this posterior distribution is proportional to the product of the model likelihood and the initial prior distribution as

$$P(p|\mathrm{H}, ..., \mathrm{T}) \propto P(\mathrm{H}, ..., \mathrm{T}|p)P(p). \tag{A.8}$$

This form ensures that when no data is observed the posterior distribution is equal to the prior assumption $P(p)$. As flips are recorded the influence of the prior distribution wanes as the posterior is dominated by the model likelihood $P(\mathrm{H}, ..., \mathrm{T}|p)$. Figure A.4 demonstrates this shift in the posterior distribution (a) from the prior (c) to the likelihood (b) after making 0, 10, and 100 observations at a ratio of 40% heads. As more observations are made with an empirical probability of $p = 0.4$, our posterior distribution becomes increasingly concentrated around that belief. We must however observe sufficient evidence to overcome the strength of our initial assumptions before coming to that conclusion with certainty.

The width and form of the posterior distribution give a full picture of the uncertainty of our the inference. Although the posterior distribution in Figure A.4a is tightly concentrated around a probability $p \sim 0.4$ after observing 100 flips, we retain some ambiguity if the parameter might deviate slightly from that peak. Nonetheless it is often useful to report the single choice of parameter most favored by the posterior distribution, known as the *maximum a posteriori* (MAP) estimate. For our example and choice of prior Eq. (A.7) this is

$$\hat{p}_{\mathrm{MAP}} = \frac{n_H + 20}{n + 40}. \tag{A.9}$$

The earlier interplay between the prior and likelihood is present in this estimate. When no coin flips are observed we obtain the prior assumption $\hat{p}_{\mathrm{MAP}} = 0.5$. As the number of observations n increases, this Bayesian estimate approaches the frequentist maximum likelihood estimate, $\hat{p}_{\mathrm{MAP}} \to \frac{n_H}{n} = \hat{p}_{\mathrm{ML}}$.

To complete Bayes' law and fully specify the posterior distribution we normalize Eq. (A.8) as

$$P(p|\mathrm{H}, ..., \mathrm{T}) = \frac{P(\mathrm{H}, ..., \mathrm{T}|p)P(p)}{P(\mathrm{H}, ..., \mathrm{T})} \tag{A.10}$$

where

$$P(\mathrm{H}, ..., \mathrm{T}) = \int_0^1 P(\mathrm{H}, ..., \mathrm{T}|p)P(p)dp \tag{A.11}$$

is the *evidence* (or "marginal likelihood") of the model. This evidence can be interpreted as the probability of arriving at the observations H, ..., T through the two stage process of first choosing a parameter p from the prior $P(p)$ then generating the data from the model likelihood $P(\mathrm{H}, ..., \mathrm{T}|p)$. By integrating over all possible parameters p, the Bayesian evidence is equal to the total probability that a model generates the observed data from our prior assumptions. A cartoon of this generative process is given in Figure A.5.

This is a useful interpretation for comparing competing models of a data set to perform *model selection*. A model with higher Bayesian evidence is more likely to have produced the observed data, and therefore has reason to be preferred. As an application we can compare the strict fair coin model described in Eq. (A.1) to the Bayesian model with the more permissive prior Eq. (A.7) on the observations (A.5) of 4 out of 10 heads. The fair coin model assumes that the coin is exactly fair regardless of the observed evidence, an

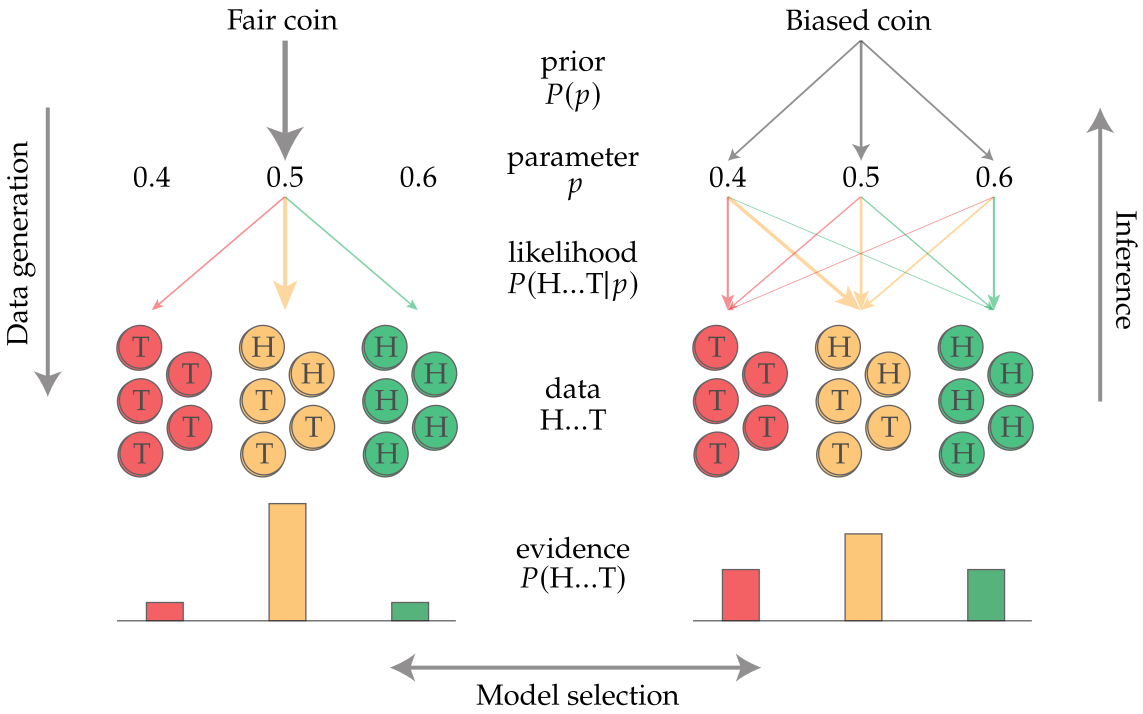


Figure A.5: Schematic of the full generative process of the fair and biased coin models. A parameter value p is first drawn from the appropriate prior $P(p)$, represented by the thicknesses of the top gray arrows to example parameter values $p = 0.4, 0.5$, and 0.6. This choice of parameter then generates the observed data according to the model likelihood $P(\text{H}, ..., \text{T}|p)$, represented by the weights of the colored arrows to the example observations of all tails, a mixture, and all heads. Lastly we compute the model evidence of each observation by summing over all generative paths leading to that outcome. For some data the fair model performs better while for others the biased coin is preferred.

assumption that can be represented with a Dirac delta function prior

$$P_{\text{fair}}(p) = \delta(p - 0.5). \tag{A.12}$$

In this case integrating over the prior simply evaluates at $p = 0.5$ to give the model evidence

$$P_{\text{fair}}(\text{H}, \ldots, \text{T}) = \int_0^1 P(\text{H}, \ldots, \text{T}|p) P_{\text{fair}}(p) dp = \frac{1}{2^n} \approx 0.097\% \tag{A.13}$$

that the fair coin model would generate our sequence. In comparison the model evidence with the prior Eq. (A.7) is

$$P_{\text{gen}}(\text{H}, \ldots, \text{T}) = \int_0^1 P(\text{H}, \ldots, \text{T}|p) P(p) dp = \frac{41!(20 + n_H)!(20 + n_T)!}{(41 + n)!20!20!} \approx 0.091\%, \tag{A.14}$$

and so this more general model is overall (slightly) less likely to have generated our observations. The ratio of such evidences is known as the *Bayes factor* between two models. Here the ratio

$$\frac{P_{\text{gen}}\overbrace{(\text{H}, \ldots, \text{T})}^{\text{10 flips}}}{P_{\text{fair}}(\text{H}, \ldots, \text{T})} \approx 0.93 \tag{A.15}$$

less than 1 indicates that the more complex model is not justified over the fair coin. Note that the Bayesian evidence has naturally penalized over-parametrization. Although the choice of parameter $p = 0.4$ does a better job than the fair coin in isolation, it has a higher model likelihood, this peak must be weighed against all possible other values of the parameter which in this case offset that advantage.

Had we instead observed 40 heads out of 100 flips, the Bayes factor flips to

$$\frac{P_{\text{gen}}\overbrace{(\text{H}, \ldots, \text{T})}^{\text{100 flips}}}{P_{\text{fair}}(\text{H}, \ldots, \text{T})} \approx 2.25 \tag{A.16}$$

as the more flexible model is now preferred in the presence of increased evidence that the coin is not quite fair. Depending on the particular data set being considered we may prefer one model or the other.

In fact, given two models M_1 and M_2, there will always be data sets that prefer one model over the other. No model can be strictly better than another across all possible observations. To see this, consider the Bayesian evidences $P_1(\boldsymbol{s})$ and $P_2(\boldsymbol{s})$ of the two models. Both of these are normalized distributions over the possible observations that a model can generate. If we suppose that the Bayesian evidence $P_1(\boldsymbol{s})$ is greater than $P_2(\boldsymbol{s})$ for all observations $\boldsymbol{s}$, both distributions can not be simultaneously normalized to 1 as

$$1 = \sum_{\boldsymbol{s}} P_1(\boldsymbol{s}) > \sum_{\boldsymbol{s}} P_2(\boldsymbol{s}) \neq 1. \tag{A.17}$$

There therefore exists both an observation $\boldsymbol{s}_1$ that favors model 1 and an observation $\boldsymbol{s}_2$ that favors model 2. In this spirit, this inherent trade-off is sometimes known as the "no free lunch" theorem [104]. When building models and performing model selection between them, we hope that "realistic data sets" fall within the preferred domain of our models.

When considering a data set, we can also compute Bayes factors like Eq. (A.16) by taking advantage of the nested structure of our model. Since the general model is equivalent to the fair coin model at the value $p = 0.5$, the Bayes factor can be written in terms of the posterior distribution at that value. Explicitly we have

$$P(p = 0.5|\mathrm{H}, ..., \mathrm{T}) = \frac{P(\mathrm{H}, ..., \mathrm{T}|p = 0.5)P(p = 0.5)}{P(\mathrm{H}, ..., \mathrm{T})} = \frac{P_{\mathrm{fair}}(\mathrm{H}, ..., \mathrm{T})}{P_{\mathrm{gen}}(\mathrm{H}, ..., \mathrm{T})}. \tag{A.18}$$

Because of this, if our posterior distribution in the general model is meaningfully peaked at $p = 0.5$, the evidence of the fair coin is greater than that of the general coin. If the posterior excludes $p = 0.5$, the reverse is true.

This formulation is useful since it is often difficult in practice to compute the absolute Bayesian evidence of a model as it involves integrating over the high dimensional space of all possible parameter values. If we define a nested model, however, we can approximate the posterior distribution using Monte Carlo methods discussed in Section A.2 relatively easily and so compute these Bayes factors and compare models.

We apply this central trick to a range of cases across Chapters 2 and 4. In each example we start with a simple model then generalize it within this Bayesian framework. By doing this we can check both whether this generalization is warranted and, if so, measure the size of the new effect, just like we infer p in the biased coin model. By stating and incorporating our prior expectations into the inference process, a Bayesian framework enables us to handle uncertainty and validate our models in a graceful manner.

A.2 Statistical physics

The earlier picture of probability and inference can be usefully cast in the language of physics and energy. By leveraging this correspondence, we can import analytic and computational tools from physics to tackle statistical questions. In this section, we will define a physical model of coin flips and demonstrate its equivalence to the statistical model discussed earlier. We will then explore how this perspective motivates alternative types of models and inference techniques.

Physics often describes systems in terms of a *configuration space* of possible *states*. In the coin flip example, the state of the system (or "configuration") is the sequence of n observed outcomes. We represent this configuration as a vector $\boldsymbol{s}$, where each entry $s_i = 1$ indicates that flip $i = 1, ..., n$ landed heads, and $s_i = 0$ represents tails. The configuration space of coin flips is thus comprised of all possible sequences of n flips, forming all binary vectors of length n.

This notation elicits the *Ising model* fundamental to statistical mechanics. The configurations of this model consist of atomic "spins" that reside at one of n "sites," each in either its *excited state* $s_i = 1$ or *ground state* $s_i = 0$[1]. We assign the excited state an *energy* of 1 and the ground state energy 0, as shown in Figure A.6. The total energy of the system, referred to as the *Hamiltonian*, is then the number of excited states (or heads)

$$H(\boldsymbol{s}) = \sum_{i=1}^{n} \left(0\delta_{s_i 0} + 1\delta_{s_i 1}\right) = n_H. \tag{A.19}$$

A more general Ising model would include coupling energies between neighboring spins, but we omit them here to maintain independent coin flips. In Appendix A.5 we discuss the deep connections between the more general interacting Ising model and statistical models of network group structure.

In an isolated system the energy Eq. (A.19) is always conserved, a fundamental law of physics. Although changes in individual spin states can occur, they must be counterbalanced by changes in other spins to maintain the "global" energy of the system. Let E be this constant energy of the system, so that only configurations $\boldsymbol{s}$ where $H(\boldsymbol{s}) = E$ are permitted, cases with $n_H = E$ excited states. The number of possible system configurations (also known as *microstates*) that satisfy this condition is then the number of ways to

[1] The two values of an Ising model spin are typically denoted as $s_i = 1$ for the "up" state and $s_i = -1$ for the "down" state, in line with the magnetic dipole moments they physically represent, although the choice of basis does not alter the underlying physics.

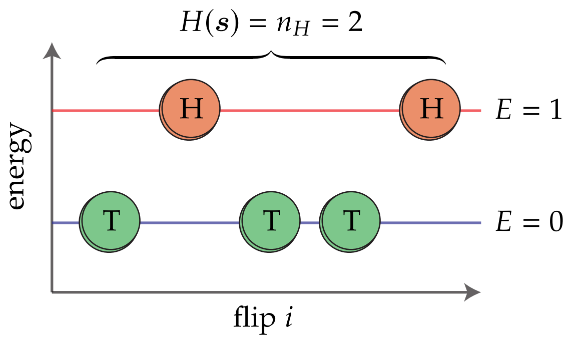


Figure A.6: Physical interpretation of coin flips. Heads ("H") have energy $E = 1$ while tails ("T") have energy $E = 0$, so that the overall energy of the configuration $\boldsymbol{s}$ is equal to the number of observed heads, $H(\boldsymbol{s}) = n_H$.

arrange the $n_H = E$ excited states among the n sites,

$$\Omega(E) = \binom{n}{E}. \tag{A.20}$$

Statistical mechanics fundamentally postulates that when a system is in *equilibrium*, all these microstates of the same total energy are equally likely to appear. This assumption defines equilibrium statistical mechanics, the framework we adopt in this thesis. Under this postulate the probability of observing any specific configuration $\boldsymbol{s}$ is therefore

$$P(\boldsymbol{s}|E) = \binom{n}{E}^{-1}. \tag{A.21}$$

This uniform distribution over all configurations of a fixed energy is known as the *microcanonical ensemble*. As a distribution over possible sequences of coin flips, it can also be interpreted as a *microcanonical model* where the continuous local probability of heads p has been replaced by the discrete global number of heads $n_H = E$.

Many of the statistical models we consider in this thesis are framed in this microcanonical form, where the "parameters" are globally observed quantities. While this formulation may not be well-suited for the coin flip example, where we have little reason to expect a "conservation of heads," it arises in other statistical settings where certain properties are conserved across random realizations. For instance, in a network of sports matches the total number of games played each season remains constant each year, even though the pattern of connections among teams changes. In Section 1.2, we explore further examples and observe how many network models can be written microcanonically.

To return to the independent coin flip model we must broaden the physical picture.

Realistically most physical systems are not truly isolated but rather exchange energy with their environment in a setting known as the *canonical ensemble*. In physical language the original system is now a *subsystem* in equilibrium with a large *thermal bath*. Although the combined system of both the subsystem and the thermal bath must still conserve overall energy, our subsystem of interest can gain and lose energy to the thermal bath.

Figure A.7 illustrates this set up for the coin flip example. Let $\tilde{s}$ represent the configuration of the thermal bath, containing $\tilde{n}$ spins and energy $\tilde{E} = H(\tilde{s})$ which counts the number of excited states in the bath. Including our subsystem $\boldsymbol{s}$, the full system $(\boldsymbol{s}, \tilde{\boldsymbol{s}})$ then has $n + \tilde{n}$ total spins and total energy $E_T = H(\boldsymbol{s}, \tilde{\boldsymbol{s}}) = E + \tilde{E}$. Since this total energy is conserved, the full system $(\boldsymbol{s}, \boldsymbol{s}')$ is uniformly distributed according to the microcanonical ensemble Eq. (A.21) of E_T excited states on $n + \tilde{n}$ sites,

$$P(\boldsymbol{s}, \tilde{\boldsymbol{s}}|E_T) = \binom{n + \tilde{n}}{E_T}^{-1}. \tag{A.22}$$

Although each pair $(\boldsymbol{s}, \boldsymbol{s}')$ with total energy E_T is equally likely, certain subsystem energies E are more likely than others. Figure A.7 demonstrates this effect with two example configurations. In panel (a) the excited states ("H") are all contained within the thermal bath $\tilde{s}$, meaning that the subsystem $\boldsymbol{s}$ has its lowest possible energy $E = 0$ and the thermal bath has the full energy E_T. In panel (b) the excited states are evenly distributed between the subsystem and the bath, and the subsystem has absorbed some energy from the bath. Across all possible configurations of the joint subsystem-bath system, imbalanced distributions like (a) are less common than balanced ones (b). Random configurations of the overall ensemble are therefore likely to be balanced. As a result, if our subsystem begins in its ground state (a) it will tend to *thermalize* into a configuration like (b) with evenly distributed excited states.

To quantify this tendency, we can count the number of overall subsystem-bath configurations $(\boldsymbol{s}, \tilde{\boldsymbol{s}})$ with subsystem energy E. As in Eq. (A.20), there are $\Omega(E)$ microstates of the subsystem $\boldsymbol{s}$ all share the same energy E. We refer to this collection of microstates as a *macrostate* of the subsystem with energy E. Figure A.8 illustrates this grouping of subsystem microstates into macrostates by energy. Macrostates that contain more microstates, those with higher $\Omega(E)$, are naturally observed more often. We can likewise construct macrostates of the thermal bath $\tilde{s}$ at energies $\tilde{E}$. Considering the number of ways to arrange the $\tilde{E}$ excited states among the $\tilde{n}$ sites of the bath, there are

$$\tilde{\Omega}(\tilde{E}) = \binom{\tilde{n}}{\tilde{E}} \tag{A.23}$$

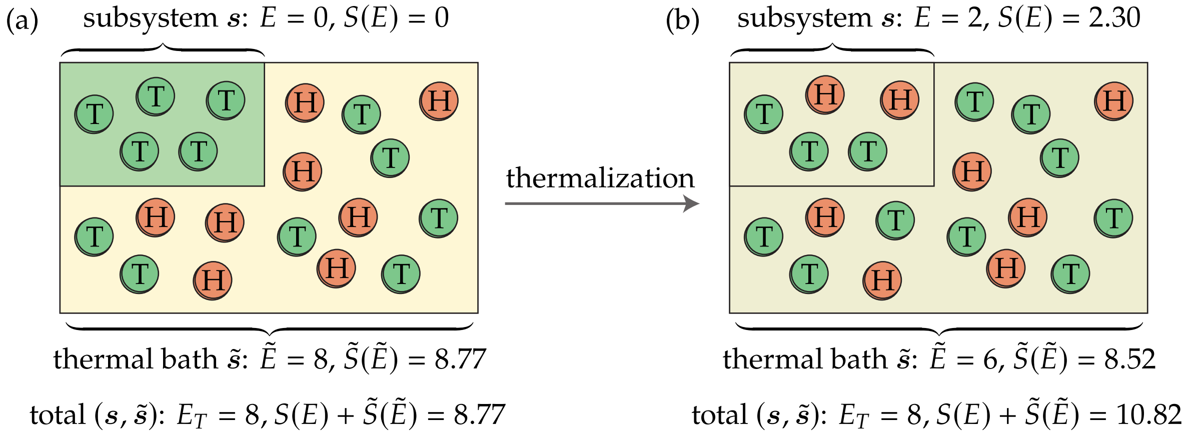

(a)
subsystem $\boldsymbol{s}$: $E = 0, S(E) = 0$
thermal bath $\tilde{\boldsymbol{s}}$: $\tilde{E} = 8, \tilde{S}(\tilde{E}) = 8.77$
total $(\boldsymbol{s}, \tilde{\boldsymbol{s}})$: $E_T = 8, S(E) + \tilde{S}(\tilde{E}) = 8.77$
thermalization
(b)
subsystem $\boldsymbol{s}$: $E = 2, S(E) = 2.30$
thermal bath $\tilde{\boldsymbol{s}}$: $\tilde{E} = 6, \tilde{S}(\tilde{E}) = 8.52$
total $(\boldsymbol{s}, \tilde{\boldsymbol{s}})$: $E_T = 8, S(E) + \tilde{S}(\tilde{E}) = 10.82$

Figure A.7: Thermalization of a subsystem (of coin flips) in contact with a thermal bath. In panel (a) all excited states ("H") are contained in the bath, an unusual configuration with overall entropy $S(E) + \tilde{S}(\tilde{E}) = 8.77$. Panel (b) has a more typical, even arrangement of the excited states, reflected in the higher total entropy $S(E) + \tilde{S}(\tilde{E}) = 10.82$. If the subsystem begins in the ground state (a) it will thus likely thermalize to the equilibrium (b).

microstates in the bath macrostate of energy $\tilde{E}$. In many settings the full microstate is not observable, for instance the precise position and velocity of each molecule of a gas. In these settings the macrostate summarizes the pieces of physically relevant information that can be observed, such as the energy or pressure.

Since the subsystem macrostate has $\Omega(E)$ microstates and the bath macrostate has $\tilde{\Omega}(\tilde{E})$, there are $\Omega(E)\tilde{\Omega}(\tilde{E})$ unique pairs $(\boldsymbol{s}, \tilde{\boldsymbol{s}})$ with subsystem energy E and bath energy $\tilde{E} = E_T - E$. Since each pair is equally likely to appear in the overall microcanonical ensemble, we thus observe subsystem energy E with probability

$$\begin{aligned} P(E|E_T) &\propto \Omega(E)\tilde{\Omega}(\tilde{E}) \\ &\propto \Omega(E)\tilde{\Omega}(E_T - E). \end{aligned} \tag{A.24}$$

These macrostate multiplicities can also be written using macrostate *entropy*, defined as the logarithm[2]

$$S(E) = \log \Omega(E) = \log \binom{n}{E}, \text{ or } \tilde{S}(\tilde{E}) = \log \tilde{\Omega}(\tilde{E}) = \log \binom{\tilde{n}}{\tilde{E}}. \tag{A.25}$$

[2] More generally defined as $S = k_B \log \Omega$ where $k_B \approx 1.38\,\mathrm{J/K}$ is the Boltzmann constant. The units of this constant are due to the thermodynamic relation Eq. (A.29).

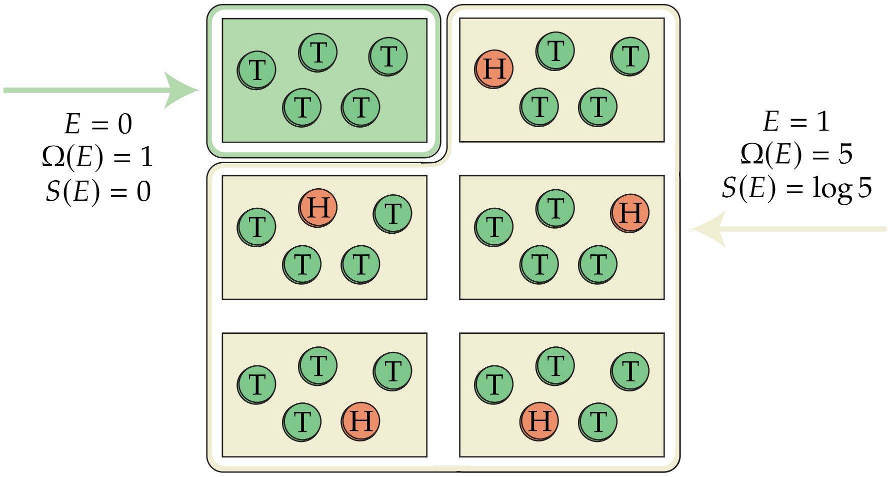


Figure A.8: Example microstates of the subsystem $\boldsymbol{s}$ grouped into macrostates by energy E. Only one microstate has $E = 0$ while five microstates share $E = 1$, meaning that the higher energy macrostate has higher entropy $S(E)$. Although each microstate is equally likely, the higher entropy macrostate is more likely to be observed.

The energy distribution then depends on the overall entropy $S(E) + \tilde{S}(\tilde{E})$ as

$$P(E|E_T) \propto e^{S(E)+\tilde{S}(\tilde{E})}. \tag{A.26}$$

Configurations with higher total entropy are therefore more likely to appear, as calculated in Figure A.7. This tendency to observe higher entropy states is the content of the 2nd law of thermodynamics: that the entropy of the universe cannot decrease. On such large scales differences in entropy are large, and the probability Eq. (A.26) approaches a certainty, a law of physics.

We can similarly compute the distribution of subsystem configurations $\boldsymbol{s}$, not just of its energy E. If we fix the subsystem microstate, valid pairs $(\boldsymbol{s}, \tilde{\boldsymbol{s}})$ correspond to bath microstates $\tilde{\boldsymbol{s}}$ among the $\tilde{\Omega}(\tilde{E})$ configurations of the remaining energy. As each pair is equally probable, the subsystem is distributed as

$$\begin{aligned} P(\boldsymbol{s}|E_T) &\propto \tilde{\Omega}(\tilde{E}) \\ &\propto e^{\tilde{S}(\tilde{E})}. \end{aligned} \tag{A.27}$$

Unlike the uniform, microcanonical ensemble of the subsystem, certain configurations are now more or less likely based on the entropy of the surrounding bath.

In this picture the *canonical* ensemble is defined by the limit $\tilde{n} \to \infty$ of a large thermal bath that can effectively absorb subsystem fluctuations. We further fix the average energy $p = \tilde{E}/\tilde{n}$ of the thermal bath in this limit, which corresponds to a fixed density of excited states. If we Stirling approximate the binomial coefficient Eq. (A.25), we find the bath's entropy $\tilde{S}(\tilde{E})$ is proportional to its energy (up to a constant C) as

$$\tilde{S}(\tilde{E}) = \beta\tilde{E} + C, \quad \beta = \log\frac{1-p}{p}. \tag{A.28}$$

This constant of proportionality β is known as the *inverse temperature* and relates changes in the bath energy $\tilde{E}$ to changes of entropy $\tilde{S}$ as

$$\frac{\partial\tilde{S}}{\partial\tilde{E}} = \beta = \frac{1}{T}, \tag{A.29}$$

aligning with the usual thermodynamic definition[3] of the temperature T. We will typically assume this temperature is positive and so added energy increases the entropy, although negative temperatures are possible in certain cases such as population inversion in laser physics or the coin flips when $p > 0.5$.

Comparing to Eq. (A.27), we then see that the probability of observing any given subsystem $\boldsymbol{s}$ in this canonical ensemble is

$$\begin{aligned} P(\boldsymbol{s}|\beta) &\propto e^{\beta\tilde{E}} \propto e^{-\beta E} \\ &\propto e^{-\beta H(\boldsymbol{s})}, \end{aligned} \tag{A.30}$$

where we have used that the total energy $E_T = E + \tilde{E}$ is conserved. In fact, by similar arguments any system $\boldsymbol{s}$ in thermal equilibrium with a large bath at inverse temperature β follows this same *Boltzmann distribution* (or "Gibbs distribution") $P(\boldsymbol{s}) \propto e^{-\beta H(\boldsymbol{s})}$, which may itself taken as the definition of the canonical ensemble of $\boldsymbol{s}$. When $T = 0, \beta \to \infty$, the subsystem is stuck in its ground state with the smallest possible energy $H(\boldsymbol{s})$. This aligns with the usual tendency of a physical system towards smaller energy, as a ball rolls down a hill. As the temperature increases and $T \to \infty, \beta = 0$, however, the Boltzmann distribution becomes uniform and every subsystem microstate is equally likely independent of its energy.

At finite $\beta > 0$ between these extremes, the typical thermal configuration may not be the ground state, even though that is the single most likely microstate. Rather, the subsystem is likely to be found in some other macrostate with higher entropy $S(E)$ as

[3] This definition ensures that if two thermal baths at temperatures $T_1 > T_2 > 0$ are brought into contact energy will flow from the higher temperature bath to the lower temperature bath to increase the overall entropy, in keeping with the 2nd law of thermodynamics.

seen in Figure A.7. Although each individual microstate of this macrostate has smaller probability than the ground state, their greater number makes their macrostate collectively more likely to be observed. This effect can be quantified by revisiting Eq. (A.26) for the distribution of energy E. Since the large bath's entropy is proportional to its energy we have

$$\begin{aligned} P(E|E_T) &\propto e^{S(E)+\tilde{S}(\tilde{E})} \propto e^{S(E)+\beta\tilde{E}} \\ &\propto e^{S(E)-\beta E} \propto e^{-\beta F(E)} \end{aligned} \tag{A.31}$$

where we have defined the *free energy*

$$F(E) = E - TS(E). \tag{A.32}$$

The most likely energy E to be observed is therefore the minimum of this free energy. Depending on the temperature T, this may no longer be the ground state energy due to the influence of the entropy $S(E)$.

To complete the description of the canonical ensemble, we normalize the Boltzmann distribution as

$$P(\boldsymbol{s}|\beta) = \frac{1}{Z(\beta)} e^{-\beta H(\boldsymbol{s})}, \quad Z(\beta) = \sum_{\boldsymbol{s}} e^{-\beta H(\boldsymbol{s})} \tag{A.33}$$

with the partition function $Z(\beta)$, a quantity which tells us much about the system in its own right. For example, its logarithmic derivative

$$\begin{aligned} -\partial_\beta \log Z(\beta) &= -\frac{1}{Z(\beta)} \partial_\beta \sum_{\boldsymbol{s}} e^{-\beta H(\boldsymbol{s})} \\ &= \frac{1}{Z(\beta)} \sum_{\boldsymbol{s}} H(\boldsymbol{s}) e^{-\beta H(\boldsymbol{s})} \\ &= \langle E \rangle \end{aligned} \tag{A.34}$$

yields the average energy $\langle E \rangle$ under the Boltzmann distribution.

If we normalize the Boltzmann distribution of our coin flip system with the partition function $Z(\beta) = (1-p)^{-n}$, we obtain

$$\begin{aligned} P(\boldsymbol{s}|p) &= \frac{1}{Z(\beta)} e^{-\beta H(\boldsymbol{s})} \\ &= \frac{1}{(1-p)^{-n}} e^{-\log\left(\frac{1-p}{p}\right) n_H} \\ &= p^{n_H}(1-p)^{n-n_H} \end{aligned} \tag{A.35}$$

which we recognize as the model likelihood Eq. (A.4) for biased coin flips with probability p. In this correspondence, we can check that the average energy relation Eq. (A.34) of the physical system indeed recovers the expected number of heads

$$\langle E \rangle = -\partial_\beta \log Z(\beta) = pn. \tag{A.36}$$

Unlike the microcanonical ensemble, the number of observed heads $n_H = E$ can now thermally fluctuate about this expectation via exchange with its surroundings in a manner that exactly reproduces independent coin flips.

This example motivates us to draw a broader analogy between statistics and physics. In Eq. (A.35) we described a physical system that reproduces the biased coin flip likelihood, the same distribution of flip outcomes given a fixed parameter p. Often, however, we are interested in the reverse inference problem of understanding the space of likely parameter values p given a particular observation. Suppose we have a generic model $P(\boldsymbol{x}|\boldsymbol{\theta})$ of data $\boldsymbol{x}$ with parameters $\boldsymbol{\theta}$. As discussed in Appendix A.1, the posterior distribution of parameters inferred from data $\boldsymbol{x}$ is by Bayes' law

$$P(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(\boldsymbol{x})}. \tag{A.37}$$

For a fixed observation $\boldsymbol{x}$, the corresponding physical system is defined by the Hamiltonian

$$H(\boldsymbol{\theta}) = -\log P(\boldsymbol{x}|\boldsymbol{\theta}) \tag{A.38}$$

over the configuration space of possible parameters $\boldsymbol{\theta}$. If we assume a uniform prior $P(\boldsymbol{\theta}) = 1$, the posterior distribution over parameters is proportional to the Boltzmann distribution of this system at unit temperature $\beta = 1$

$$P(\boldsymbol{\theta}|\boldsymbol{x}) \propto P(\boldsymbol{x}|\boldsymbol{\theta}) \propto e^{-H(\boldsymbol{\theta})}. \tag{A.39}$$

In fact, if we normalize the Boltzmann distribution as

$$P(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{1}{Z(1)} e^{-H(\boldsymbol{\theta})}, \tag{A.40}$$

the partition function $Z(1)$ is equal to the Bayesian evidence

$$Z(1) = \sum_{\boldsymbol{\theta}} e^{-H(\boldsymbol{\theta})} = \sum_{\boldsymbol{\theta}} P(\boldsymbol{x}|\boldsymbol{\theta}) P(\boldsymbol{\theta}) = P(\boldsymbol{x}). \tag{A.41}$$

Through this correspondence we can explore the posterior distribution of any Bayesian model by simulating the behavior of the analogous physical system.

This equivalence highlights a subtle philosophical difference between the physical perspective and the common statistical view of these problems. In statistics, particularly in frequentist treatments, we often consider and report the single best-fit parameter of a model that maximizes $P(\boldsymbol{\theta}|\boldsymbol{x})$. Physically, this is akin to identifying the ground state configuration with the lowest energy $H(\boldsymbol{\theta})$.

Yet in physics, we are usually more interested in the typical behavior of the system across the entire thermal ensemble rather than the nature of the single most probable microstate. As shown in Figure A.7, there may be many other less likely yet more entropic configurations that dominate the overall distribution when taken together. In common glass, for example, although the ground state would be an ordered crystalline structure, thermal fluctuations make the typical configuration amorphous, giving the material its signature uniform transparency. Keeping with this perspective, in this work we will consider the full posterior distribution when possible to give a comprehensive picture of system behavior.

For this purpose we employ Markov Chain Monte Carlo (MCMC) methods, a common technique to simulate generic physical systems and so to sample from generic probability distributions. This strategy performs a weighted random walk over configuration space, analogous to thermalization and dispersion in a real system. Returning to our coin flip example, we can consider how in Figure A.7 the subsystem dynamically evolves. If our subsystem $\boldsymbol{s}$ begins in a the low entropy ground state (a), through the jostling of the coins the system will naturally tend towards a typical, high entropy configuration (b). Yet physically this transition is not instantaneous. For instance, gas molecules kinetically bump into each other and gradually disperse throughout their enclosure. We use Markov chain methods to simulate coin flip thermalization as a process that occurs one coin flip at a time.

A Markov chain walks through configuration space in discrete increments of time. At each step the Markov chain *transitions* from one state at time t to another state at slightly later time $t + \Delta t$. This is a random walk, where a move from state $\boldsymbol{s}$ to $\boldsymbol{s}'$ is made with probability $P(\boldsymbol{s} \rightarrow \boldsymbol{s}')$. Under mild conditions on the transition probabilities, this Markov chain will converge to some equilibrium distribution $P(\boldsymbol{s})$. If the states of the Markov chain are distributed according the equilibrium $P(\boldsymbol{s})$, the final $\boldsymbol{s}'$ after the chain move $\boldsymbol{s} \rightarrow \boldsymbol{s}'$ will by definition share the equilibrium distribution

$$P(\boldsymbol{s}') = \sum_{\boldsymbol{s}} P(\boldsymbol{s})P(\boldsymbol{s} \rightarrow \boldsymbol{s}'). \tag{A.42}$$

We are then left with the problem of constructing a Markov chain which has the particular equilibrium distribution $P(\boldsymbol{s})$ we are interested in.

A simple and ubiquitous way to establish such a chain is known as the *Metropolis-Hastings algorithm*. Each step in this Markov chain consists of two parts. First, a move to a new state is *proposed* given the current state. Second, that move is either *accepted* and the chain moves to the new state, or it is *rejected* and the chain remains at its current state. If we propose moves with transition probabilities $P_{\text{prop}}(\boldsymbol{s} \rightarrow \boldsymbol{s}')$, we accept each proposal with probability

$$P_{\text{acc}}(\boldsymbol{s} \rightarrow \boldsymbol{s}') = \min\left(1, \frac{P(\boldsymbol{s}')P_{\text{prop}}(\boldsymbol{s}' \rightarrow \boldsymbol{s})}{P(\boldsymbol{s})P_{\text{prop}}(\boldsymbol{s} \rightarrow \boldsymbol{s}')}\right). \tag{A.43}$$

We will often use symmetric proposals where $P_{\text{prop}}(\boldsymbol{s} \rightarrow \boldsymbol{s}') = P_{\text{prop}}(\boldsymbol{s}' \rightarrow \boldsymbol{s})$, for which this acceptance probability simplifies to

$$P_{\text{acc}}(\boldsymbol{s} \rightarrow \boldsymbol{s}') = \min\left(1, \frac{P(\boldsymbol{s}')}{P(\boldsymbol{s})}\right) = \min\left(1, e^{\beta(H(\boldsymbol{s})-H(\boldsymbol{s}'))}\right). \tag{A.44}$$

In terms of the system energy $H(\boldsymbol{s})$, we see that the algorithm will always accept changes that decrease the energy, and occasionally moves that increase it, an emulation of how the real system evolves.

From this two-step process the overall Markov chain transition probabilities are then distinguished by if they increase or decrease the probability as

$$P(\boldsymbol{s} \rightarrow \boldsymbol{s}') = \begin{cases} P_{\text{prop}}(\boldsymbol{s} \rightarrow \boldsymbol{s}') & P(\boldsymbol{s}') \geq P(\boldsymbol{s}), \boldsymbol{s} \neq \boldsymbol{s}' \\ P_{\text{prop}}(\boldsymbol{s} \rightarrow \boldsymbol{s}')\frac{P(\boldsymbol{s}')}{P(\boldsymbol{s})} & P(\boldsymbol{s}') < P(\boldsymbol{s}) \\ P_{\text{prop}}(\boldsymbol{s} \rightarrow \boldsymbol{s}) + \sum_{\boldsymbol{s}'<\boldsymbol{s}} P_{\text{prop}}(\boldsymbol{s} \rightarrow \boldsymbol{s}')\left[1 - \frac{P(\boldsymbol{s}')}{P(\boldsymbol{s})}\right] & \boldsymbol{s}' = \boldsymbol{s} \end{cases} \tag{A.45}$$

where sum in the case $s' = s$ accounts for all the rejected proposals to states s' with probability $P(s') < P(s)$. Summing over these cases, now of states s that can produce a state s', we can check the stability condition Eq. (A.42) as

$$\begin{aligned}\sum_{s} P(s)P(s \to s') &= \sum_{s<s'} P(s)P_{\text{prop}}(s \to s') + \sum_{s>s'} P(s)\frac{P(s')}{P(s)}P_{\text{prop}}(s \to s') \\ &\quad + P(s')P_{\text{prop}}(s' \to s') + P(s')\sum_{s>s'} P_{\text{prop}}(s' \to s)\left[1 - \frac{P(s)}{P(s')}\right] \\ &= P(s')\sum_{s} P_{\text{prop}}(s \to s') = P(s'). \end{aligned} \tag{A.46}$$

Therefore the desired distribution is a fixed point of the Metropolis-Hastings algorithm, regardless of the choice of proposals $P_{\text{prop}}(s \to s')$. However, being a Markov chain, there is still a clear correlation between subsequent steps in the random walk. Only after a typical number of Markov chain known as the *mixing time* are the samples meaningfully independent. Therefore to efficiently obtain independent samples from the distribution, to for example compute expectations, the mixing time should be as small as possible. The form these proposals take can considerably impact the mixing time, and clever choices are often needed to make Monte Carlo methods tractable.

A common choice of proposal for discrete settings like this is to consider *single site flips* where we choose a random site i and flip it from heads to tails, $s_i = 1 \mapsto 0$ or vice versa. An advantage to this local change is that the resulting chance in the probability (or energy) is small and therefore likely to be accepted. If a entirely new global configuration is drawn uniformly at random, it likely has a much lower probability (or higher energy) than the current sample, and so will likely be rejected and waste an iteration of the algorithm.

If we apply this to our simple coin flipping example, Figure A.9 shows how the number of flips changes over the course of the Metropolis-Hastings algorithm. We can observe that although the system starts in a configuration not particularly representative of the equilibrium distribution, after many iterations the Markov chain thermalizes the state. Although this is a fairly simple distribution, one we could sample directly, the flexibility of MCMC allows us apply it to far more complex distributions to draw inferences. In doing this, however, we must be careful to ensure that the Markov chain has adequately converged. If samples are interpreted too early in the algorithm, results will be skewed by the initial state, as in the initial samples of Figure A.9.

These Monte Carlo methods are very powerful, and are the workhorse of the statistical inferences made in this thesis. In order to apply these methods to their fullest, we can further augment the Metropolis-Hastings algorithm by performing *parallel tempering* to

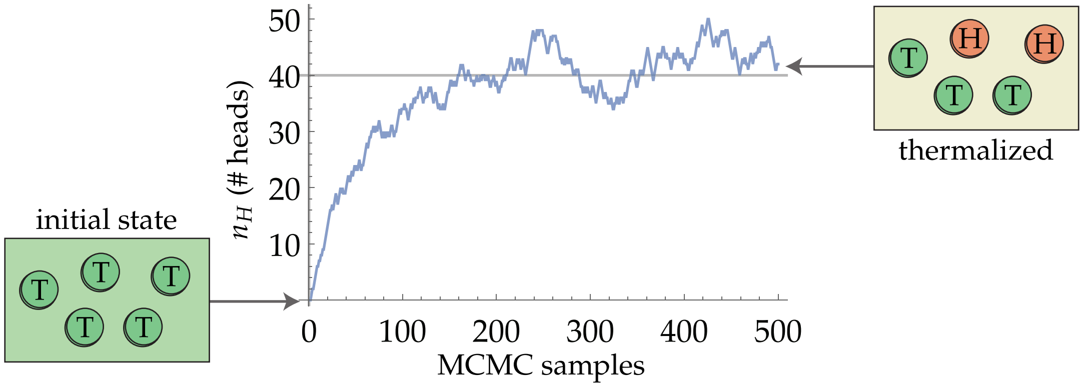


Figure A.9: Trajectory of the Metropolis-Hastings MCMC algorithm to thermalize a system of 100 coin flips. The system begins in the ground state where $n_H = 0$. The average value $n_H = 40$ is highlighted. Once the Markov chain has thermalized, the samples are indicative of this average value.

reduce the mixing time of the Markov chains and *thermodynamic integration* to compute the Bayesian evidence in a manner analogous to how such methods are used to compute the physical free energy.

A.3 Information theory

In the previous section we discussed how a physical perspective on statistics naturally leads to concepts like entropy and algorithms for exploring configuration spaces. In this section we discuss an information theoretic perspective which abstractly considers the sequence of coin flips as a message to be transmitted, allowing us to reckon with the inherent complexity of the data observed in a general sense.

Central to information theory is a thought experiment aiming to *encode* a message as efficiently as possible. Suppose we would like to transmit to another party the results of our earlier experiment of 10 coin flips as the message "HHTTHTTHTT." Suppose further that we are restricted in this communication to use a *binary channel* which can only send a binary sequence of 0's and 1's of our choosing. We would then like the receiver on the other end of the channel to be able to decode our binary transmission back into our original message.

Just as we represented the sequence as a physical state vector in the last section, we may now encode these coin flips as the binary string "1100100100" where the digit "1" represents heads and "0" represents tails. This correspondence between meanings and binary strings is known as a *codebook*. So long as we and the receiver agree on the nature of the encoding, the receiver can use the codebook to decode the binary string back into the original message of outcomes. To determine the efficiency of our transmission we measure the length of our binary string in *bits*. In this case our encoding used 10 digits (bits) to transmit the message, one for each flip. A schematic of this encoding framework is given in Figure A.1c.

This association of digits to outcomes is a natural encoding of two-sided coin flips. In a more general setting, however, we will need to be more creative in our transmission. Suppose we would instead like to send the message "ELEVENELVES" as a binary string. Since this string contains 5 distinct characters, we can no longer encode the message by assigning each character its own binary digit. Some characters must be represented by a *codeword* of multiple binary digits. If we are not careful, however, this can render our message ambiguous. If we assign "E" to the codeword "0" and "L" to the codeword "00," the binary message "00" could decode into either "EE" or "L."

To avoid this polysemy, we can ensure that our transmission is uniquely decodable by using a *prefix-free* (or "instantaneous") code. If no codeword in our codebook is a prefix of another codeword, as the receiver reads the message from left to right the divisions between the codewords will always be clear. Our earlier example was not prefix-free as the codeword "0" is a prefix of the codeword "00," leading to the double meaning.

Any such prefix-free binary code can be usefully represented as a binary tree whose leaves each correspond to a character (or "symbol") being transmitted. The codeword associated to each symbol is then represented by the path from the root of the tree down to that symbol. Figure A.10a shows an example of such a tree used to encode the five characters. Following the tree paths this "balanced" encoding represents the characters {"E", "L", "V", "N", "S"} with the codewords {"00", "01", "10", "110", "111"} respectively. With this codebook we can then encode the phrase "ELEVENELVES" into pictured binary string of length 24 bits and ensure that it uniquely decodes back into our desired dispatch.

Now, a key goal of information theory is not only to successfully transmit a message but also to do so using as few bits as possible. This objective can be seen as a formalization of Occam's razor, the scientific principle that favors the simplest possible answer to a question. In this analogy, transmitting our binary string effectively "explains" to the receiver the data we have observed, making a shorter transmission a more succinct explanation. If we understand predictable patterns in our observations, we can exploit them to construct a more efficient encodings. There is a fundamental duality between *compression* and modeling, as in this context, to compress is to understand.

In this spirit we can look for patterns in our message to try and come up with a clever way to shorten our encoding. The phrase "ELEVENELVES" has a rather lopsided distribution

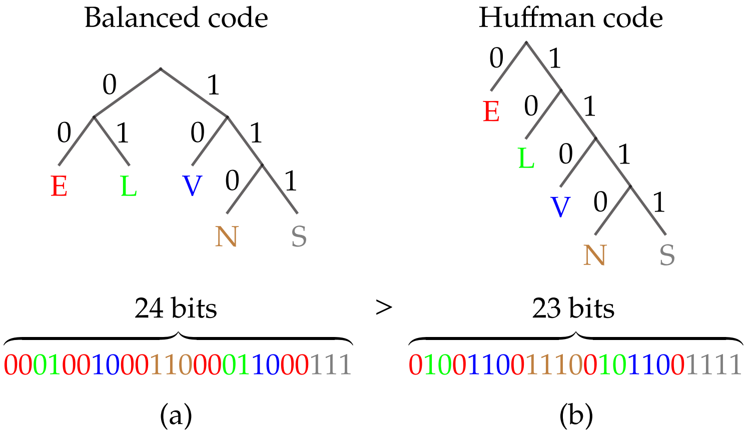


Figure A.10: Examples of an (a) balanced code and (b) optimized Huffman code to convert the phrase "ELEVENELVES" into a binary string. The string associated to each letter is denoted by the path from the top of the tree down to the appropriate node. The Huffman code is able to produce a shorter overall message than the balanced code by representing the common letter "E" with a short string "0" despite representing the uncommon letters "N" and "S" with longer strings.

of characters at 5 E's, 2 L's, 2 V's, 1 N, and 1 S. Some of our codewords are therefore being used in the transmission much more frequently than others. If symbol $r = 1, ..., q$ of the q symbols appears n_r times in our message as a codeword of length ℓ_r, the total message length is

$$\sum_{r=1}^{q} n_r \ell_r. \tag{A.47}$$

To shorten the overall message we would therefore like the codewords to be as short as possible and to prioritize shortening frequently occurring symbols. In our original encoding the symbol "E" is transmitted with the codeword "00" at a cost of two bits apiece. Since "E" appears so frequently in the message it may be wise to instead represent it with a shorter codeword like "0" and save 5 bits in our total transmission.

Yet this choice has a cost. If we assign "0" to represent "E" none of the other four characters can be represented with a codeword that begins with a "1" or else the code would no longer be prefix-free. When shortening one codeword we must necessarily lengthen other codewords. This tradeoff is the content of *Kraft's inequality*, which states that the codeword lengths of any prefix-free code must satisfy

$$\sum_{r=1}^{q} 2^{-\ell_r} \leq 1, \tag{A.48}$$

considering the fraction of the binary tree each codeword occupies. Given the frequencies n_r at which each symbol appears, we would like to select codeword lengths that minimize the total message length Eq. (A.47) subject to the prefix-free constraint Eq. (A.48).

Huffman codes strike this balance and provably minimize the message length by assigning short codewords to frequent symbols while saturating Kraft's inequality. Figure A.10b contains an example of a Huffman code for our application. The code shortens the codeword for "E" from 2 to 1 bit while lengthening the codewords for "N" and "S" from 3 to 4 bits. Since "E" appears much more frequently than "N" and "S," this change shortens the overall message from 24 to 23 bits. By adapting the encoding to the nature of the data, the Huffman code achieves a more parsimonious representation of the message.

More generally given a frequency distribution $\{n_r\}$ of symbols one can always construct a Huffman code with a simple recursive algorithm. The resulting optimal codeword lengths $\{\ell_r\}$ follow a predictable pattern. If a symbol appears at a fraction $p_r = \frac{n_r}{n}$ among the n total symbols, the length of its associated codeword satisfies

$$-\log_2(p_r) \leq \ell_r < -\log_2(p_r) + 1. \tag{A.49}$$

Frequent symbols with high probability p_r are thus assigned small codewords as $-\log_2(p_r)$ is small while infrequent symbols use longer codewords. In our "ELEVENELVES" example, the character "E" appears with probability $p = 5/11$, and is so assigned a string of length $1 \approx \log_2(11/5)$ while the character "S" appears at the ratio $1/11$ and is encoded using $4 \approx \log_2(11)$ bits.

In most contexts we consider, symbol probabilities are small and codeword lengths are long. In this regime we can approximate lengths as $\ell_r = -\log_2(p_r)$, which would in fact be the optimal choices if the lengths could be non-integral numbers of bits. Using these optimal codeword lengths, the minimum message length per symbol is

$$S[\{p_r\}] = \frac{1}{n}\sum_{r=1}^{q} n_r \ell_r = -\sum_{r=1}^{q} p_r \log_2 p_r, \tag{A.50}$$

known as the *Shannon entropy* (or simply "entropy") of the distribution $\{p_r\}$. For continuous distributions $P(\boldsymbol{x})$ this entropy generalizes to

$$S[P] = -\int P(\boldsymbol{x}) \log_2 P(\boldsymbol{x}) d\boldsymbol{x}. \tag{A.51}$$

By providing an information theoretic lower bound on transmission, the Shannon entropy captures the inherent information content of a probability distribution that no amount of clever encoding tricks can overcome.

We can make contact between this information-theoretic entropy and the physical entropy described in Section A.2. There the microcanonical ensemble is the uniform distribution $P(\boldsymbol{s}) = \frac{1}{\Omega}$ over all possible configurations that conserve the total energy. The Shannon entropy Eq. (A.50) then agrees which the microcanonical entropy $S = \log \Omega$ of Eq. (**??**). From this perspective, a macrostate with high physical entropy is one where a large amount of information is required to specify which of the many possible microstates it represents.

We also note that the uniform distribution has the highest entropy among all possible distributions $\{p_r\}$ on q objects. By the convexity of the logarithm,

$$S[\{p_r\}] = -\sum_{r=1}^{q} p_r \log p_r \leq -\sum_{r=1}^{q} \frac{1}{q} \log\left(\frac{1}{q}\right) = \log q. \tag{A.52}$$

This observation gives another motivation for the microcanonical ensemble: the *maximum-entropy* distribution over possible configurations. Since the entropy measures how structured, how compressible, a probability distribution is, this maximum-entropy distribution is structureless and maximally agnostic: a reasonable properties of an equilibrium distribution where any initial structure is thermalized away.

The canonical ensemble $P(\boldsymbol{s}) \propto e^{-\beta H(\boldsymbol{s})}$ can likewise be motivated as a maximum entropy distribution with a given average energy, which determines the choice of β. When designing priors for Bayesian inference, we will frequently appeal to this minimally assumptive principle and choose maximum-entropy priors subject to certain constraints we expect the system to provide. For example, a Gaussian distribution can be motivated as the maximum entropy distribution of a real random variable of a fixed mean and variance.

Returning to encodings, to obtain the entropy we had considered the total message length of a Huffman code optimized to send that particular message. If we believe that the symbols will be distributed with probabilities $\{q_r\}$ we should optimize our Huffman code accordingly to have code lengths $\ell_r = -\log q_r$. However in many contexts we do not *a priori* know what distribution of symbols to expect. We may assume a distribution $\{q_r\}$ that is not borne out in practice. If the symbols we must transmit have true probabilities $\{p_r\}$ the average code length becomes

$$\sum_r p_r(-\log q_r). \tag{A.53}$$

Had we used code lengths $-\log p_r$ attuned to the true distribution, this would give the Shannon lower bound. Since the our encoding is *misspecified* it will instead require a larger number of bits to transmit. The shortfall between the two, the extra cost we incur, is known as the *Kullback-Leibler (KL) divergence* between the true distribution $\{p_r\}$ and our assumption $\{q_r\}$:

$$\begin{aligned} D_{\mathrm{KL}}(\{p_r\}||\{q_r\}) &= \left(\sum_{k=1}^{q} p_r(-\log q_r)\right) - \left(\sum_{k=1}^{q} p_r(-\log p_r)\right) \\ &= \sum_{k=1}^{q} p_r \log \frac{p_r}{q_r} \geq 0. \end{aligned} \tag{A.54}$$

This story repeats when modeling data. Given a model with distribution $Q(\boldsymbol{x})$ we can write the Huffman code length (or *description length*) of an observation $\boldsymbol{x}$ as

$$H(\boldsymbol{x}) = -\log Q(\boldsymbol{x}), \tag{A.55}$$

which we can compare to the Hamiltonian in Eq. (A.38). If we use this model encoding on a stream of observations whose true distribution is $P(\boldsymbol{x})$, the average description length decomposes as

$$\begin{aligned} \sum_{\boldsymbol{x}} P(\boldsymbol{x})H(\boldsymbol{x}) &= -\sum_{\boldsymbol{x}} P(\boldsymbol{x})\log P(\boldsymbol{x}) + \sum_{\boldsymbol{x}} P(\boldsymbol{x})\log\frac{P(\boldsymbol{x})}{Q(\boldsymbol{x})} \\ &= \underbrace{S[P]}_{\text{entropy}} + \underbrace{D_{\mathrm{KL}}(P||Q)}_{\text{cross-entropy}} \end{aligned} \tag{A.56}$$

into the inherent entropy of the data and the *cross-entropy* cost of our model misspecification. As we model the random process $P(\boldsymbol{x})$ our average description length can never fall below the entropic lower bound, but any description length above this point is evidence of the failure of our model to match reality. While it is relatively straightforward to measure this average description length in practice, deducing what fraction of it is due to the entropy or the cross-entropy is a hard problem. When comparing the average description lengths of two models on the same stream of data, however, we can confidently attribute their difference to a difference in the cross-entropies and prefer the model with the smaller average description length. This motivates the *minimum description length* (MDL) principle, which prefers models whose corresponding encodings across realistic data sets are as small as possible.

As an application, suppose that we would like to select the appropriate value of a parameter $\boldsymbol{\theta}$ for a model $P(\boldsymbol{x}|\boldsymbol{\theta})$. For each choice of parameter, the corresponding model description length is simply the minus log-likelihood $H(\boldsymbol{x}|\boldsymbol{\theta}) = -\log P(\boldsymbol{x}|\boldsymbol{\theta})$. Choosing the model that minimizes the description length therefore amounts to finding the maximum-likelihood estimate of the parameter as

$$\mathrm{argmin}_{\boldsymbol{\theta}} H(\boldsymbol{x}|\boldsymbol{\theta}) = \mathrm{argmax}_{\boldsymbol{\theta}} P(\boldsymbol{x}|\boldsymbol{\theta}). \tag{A.57}$$

As discussed in Section A.1, however, this maximum likelihood estimation is prone to overfitting. This approach is also problematic from an information-theoretic perspective. In our optimization of the transmission we have neglected the cost of transmitting the parameter $\boldsymbol{\theta}$ itself. In the Bayesian context this parameter will be distributed according to a prior $P(\boldsymbol{\theta})$ that corresponds to its own encoding

$$H(\boldsymbol{\theta}) = -\log P(\boldsymbol{\theta}). \tag{A.58}$$

If we consider the total information cost of this now two stage process of first transmitting the parameter $\boldsymbol{\theta}$ and then the data $\boldsymbol{x}$ given that parameter, we recover Bayesian *maximum a posteriori* (MAP) estimation

$$\begin{aligned} \mathrm{argmin}_{\boldsymbol{\theta}} \left[H(\boldsymbol{x}|\boldsymbol{\theta}) + H(\boldsymbol{\theta}) \right] &= \mathrm{argmax}_{\boldsymbol{\theta}} P(\boldsymbol{x}|\boldsymbol{\theta}) P(\boldsymbol{\theta}) \\ &= \mathrm{argmax}_{\boldsymbol{\theta}} P(\boldsymbol{\theta}|\boldsymbol{x}). \end{aligned} \tag{A.59}$$

As seen earlier, the maximum likelihood and MAP estimates of a parameter often differ considerably, particularly when a relatively small amount of data is available. In Section 3.1 on the applications of information theory to network science we will again encounter situations where neglecting certain terms of the transmission process leads to wildly different results.

After the two stage transmission process of Eq. (A.59) we transmit to the receiver both the data of interest $\boldsymbol{x}$ and the best-fit parameter $\boldsymbol{\theta}$ used. When assessing model performance, however, knowledge of the the parameter is often redundant to the data. We can instead holistically evaluate model performance using the Bayesian evidence, the probability that a model generates a particular data $\boldsymbol{x}$ summed over all possible parameters

$$P(\boldsymbol{x}) = \int P(\boldsymbol{x}|\boldsymbol{\theta})P(\boldsymbol{\theta}). \tag{A.60}$$

The description length of the integrated model is then

$$H(\boldsymbol{x}) = -\log P(\boldsymbol{x}). \tag{A.61}$$

Therefore we can motivate model selection that chooses model with higher Bayesian evidence, as when computing Bayes factors, as selecting the model that more efficiently compresses the data integrated over its latent parameters. These connections highlight the duality between the compression and modeling of a data set.

To this point, we have considered the Shannon entropy of *probability distributions*. However, much of this thesis focuses on a subtly different notion of the complexity of *objects*. Shannon entropy quantifies the complexity of a probability distribution without regard to the specific objects within that distribution. For instance, suppose that we want to transmit one of two messages: the full text of *Dune* by Frank Hebert or *It* by Stephen King. While each book's content is undoubtedly "complex," our current framework would allow us to "transmit" them with minimal information cost. If we define an encoding where "0" represents the text of *Dune* and "1" stands for *It*, we could send a single "0" to transmit the entirety of *Dune*. This setup might suggest that the inherent information cost of *Dune*'s content is just one bit, which is clearly an unreasonable conclusion.

The problem is that we have overlooked the information cost required to establish the codebooks. If the receiver is unaware of our coding scheme, we must first communicate the full text that each binary digit corresponds to before our transmission. Once this scheme is established, we can indeed send our choice of book with a single bit repeatedly at low cost. However, the initial information cost of creating the codebook is much higher. Shannon entropy measures the information needed to transmit objects drawn from a probability distribution, not the complexity of the objects themselves.

To broach the information content of an object, we should instead turn to the *Kolmogorov complexity*. Certain objects and outcomes appear to be inherently more complicated than others. For example a sequence of coin flips "HHHHHHHHHH" is easy to describe as "10 heads in a row." Even if the sequence was 1000 heads, the outcome would not be much more complex to describe. On the other hand, the pattern of coin flips "HHTTHTTHTT" we observed appears to be more complicated to describe. However, even in this case the coin flips we observed are simply the first 10 digits of $\pi = 11.00100100..._2$ in binary: a concise, if unusual, explanation. Yet if we are presented with a truly "random" string of coin flips, there is little hope for such an efficient description of the outcomes. The Kolmogorov complexity is meant to capture the difference between these settings and fundamentally measure how structured a given data set is.

Roughly speaking, the Kolmogorov complexity $K(\boldsymbol{x})$ can be understood as the length of the shortest computer program that would output the object (typically string) in question. In our earlier examples, this program might be "output 10 heads" or "first 10 digits of π_2" in pseudocode. In our earlier example of books, the full texts may be compressed with a technique like the Lempel-Ziv-Welch algorithm used in the `.gif` file format. The "program" in this case would consist of a description of the LZW algorithm, followed by the compressed file. The total program size, and so complexity will still be fairly large as some fraction of the original length of the book, but is much greater than the single bit we had used to transmit it in a probabilistic sense.

The "computer program[4]" in the definition of the Kolmogorov complexity is vague enough to accommodate any possible valid encoding or explanation of a string. For example, we can consider the Huffman encoding of the data $\boldsymbol{x}$ generated by a model M. We can imagine a computer program which consists of a description of the model M itself, then provides the Huffman code as a binary string of length $H_M(\boldsymbol{x})$ that can be decoded with knowledge of the model. When the data set is large, we typically neglect the constant overhead required to describe the model and this framework and roughly say that an encoding of length $H_M(\boldsymbol{x})$ is possible for the data $\boldsymbol{x}$[5].

From this perspective each model can be viewed as a competing encoding of the data, each of which provides an upper bound on the inherent complexity. If we have a basket of candidate models $M \in \mathscr{M}$, we can then loosely approximate the "true" information cost of $\boldsymbol{x}$ as the minimum

$$K(\boldsymbol{x}) \sim \min_{M \in \mathscr{M}} H_M(\boldsymbol{x}). \tag{A.62}$$

The higher the model evidence the shorter the description length, yielding a tighter upper bound on the true cost.

Despite this relative improvement, it is not possible to conclusively show that our approximation is particularly close to the truth. There may always be clever encoding out there that transmits the data far more efficiently than the models we consider. To show that $K(\boldsymbol{x})$ is above some value n, we would need to check the outputs of all possible

[4] More formally a universal Turing machine.

[5] In this pursuit we cannot consider models too finely attuned to a particular data set, or else we can no longer neglect the cost to specify the model itself.

programs of length less than or equal to n, an uncomputable task. For example, we may not recognize our initial sequence of coin flips "HHTTHTTHTT" as the first 10 digits of $\pi = 11.00100100..._2$ in binary, a more concise explanation that our models.

Although we cannot find the perfect encoding, nor the perfect model, we can strive for a better understanding of systems and their complexity in this information-theoretic framework.

A.4 Prediction and validation

In the previous sections, we explored perspectives on *unsupervised* machine learning tasks, which aim to understand a data set in isolation without guidance or predefined outcomes. For instance, we can infer the probability of heads from a sequence of coin flips or deduce the group structure from a network using only the pattern of connections. We also discussed how measures like Bayesian evidence or description length assess the quality of our model in an information-theoretic manner that is intrinsic to the data.

Often, however, we may be interested in applying our model and its inferences beyond the scope of the initial data set. One application is to *predict* future events; for example, in a college football network of match outcomes, we might predict the winner between two teams that did not compete during the regular season. Additionally, we may *validate* our inferences against expert knowledge or existing context in a *supervised* setting, or compare the outputs of different models applied to the same data set. Machine learning offers a variety of tools to address these practical purposes.

If we assume that the same mechanisms that generated our observed data also inform unobserved outcomes, we can leverage our model inferences to make predictions. For example, if we observe a coin and are convinced it is fair, we may predict that future flips of the coin will land heads and tails with equal probability. This extrapolation may or may not be accurate. In machine learning terminology, we initially *fit* the model to the "training" data and then assess the quality of the resulting predictions using a "testing" data set.

In our coin flip example, we may split the sequence $\boldsymbol{s}$ we observe into a training set $\boldsymbol{s}^{\text{train}}$ of n^{train} flips and testing set $\boldsymbol{s}^{\text{test}}$ of n^{test} flips. Figure A.1d provides a schematic of this *cross-validation* set up. After fitting the model to the training data we obtain the posterior distribution of the probability p, represented as $P(p|\boldsymbol{s}^{\text{train}})$, which is maximized by the best fit

$$\hat{p}_{\text{MAP}}^{\text{train}} = \frac{n_H^{\text{train}} + 20}{n^{\text{train}} + 40} \tag{A.63}$$

as in Eq.(A.9). Assuming the withheld testing data $\boldsymbol{s}^{\text{test}}$ is governed by the same parameter $\hat{p}_{\text{MAP}}^{\text{train}}$ as the *training* data, we can evaluate the likelihood Eq. (A.4) on the *testing* data

$$P(\boldsymbol{s}^{\text{test}}|\hat{p}_{\text{MAP}}^{\text{train}}) = \left(\frac{n_H^{\text{train}} + 20}{n^{\text{train}} + 40}\right)^{n_H^{\text{test}}} \left(\frac{n_T^{\text{train}} + 20}{n^{\text{train}} + 40}\right)^{n_T^{\text{test}}} . \tag{A.64}$$

This serves as a measure of the model's out-of-sample predictive performance.

While most cross-validation tests use the single best parameter, we can instead use the full posterior distribution of possible parameters to compute the *posterior predictive*

$$\begin{aligned} P(\boldsymbol{s}^{\text{test}}|\boldsymbol{s}^{\text{train}}) &= \int P(\boldsymbol{s}^{\text{test}}|p)P(p|\boldsymbol{s}^{\text{train}})dp \\ &= \frac{(n^{\text{train}} + 41)!(n_H + 20)!(n_T + 20)!}{(n + 41)!(n_H^{\text{train}} + 20)!(n_H^{\text{train}} + 20)!}. \end{aligned} \tag{A.65}$$

This distribution is equal to the probability that the model generates the data $\boldsymbol{s}^{\text{test}}$ conditioned on it also generating $\boldsymbol{s}^{\text{train}}$.

In a cross validation context, the initial data set $\boldsymbol{s}$ is randomly split into the training and testing data sets, often at a 80/20 ratio. The predictive performance of the model is quantified using either the likelihood or posterior predictive. In keeping with the information theoretic interpretation Eq. (A.58), we typically report the negative log likelihood or posterior predictive as

$$\begin{aligned} \langle H(\boldsymbol{s}^{\text{test}}|\hat{p}_{\text{MAP}}^{\text{train}})\rangle_{\boldsymbol{s}^{\text{test}},\boldsymbol{s}^{\text{train}}} &= \langle -\log P(\boldsymbol{s}^{\text{test}}|\hat{p}_{\text{MAP}}^{\text{train}})\rangle_{\boldsymbol{s}^{\text{test}},\boldsymbol{s}^{\text{train}}}, \\ \langle H(\boldsymbol{s}^{\text{test}}|\boldsymbol{s}^{\text{train}})\rangle_{\boldsymbol{s}^{\text{test}},\boldsymbol{s}^{\text{train}}} &= \langle -\log P(\boldsymbol{s}^{\text{test}}|\boldsymbol{s}^{\text{train}})\rangle_{\boldsymbol{s}^{\text{test}},\boldsymbol{s}^{\text{train}}}, \end{aligned} \tag{A.66}$$

where the results are averaged over many possible validation splits $\boldsymbol{s}^{\text{train}}$, $\boldsymbol{s}^{\text{test}}$. In practice the likelihood and posterior predictive can give different results, but we will generally prefer to use the latter to evaluate the full posterior of possible parameter values.

The Bayesian evidence can also be viewed as a measure of predictive performance, averaged over various data splits. We can write out our data set $\boldsymbol{s}$ as the sequence of coin flips $s_1, ..., s_n$. Bayesian evidence is the probability that the model generates this entire sequence. Meanwhile, the posterior predictive is the probability that the model generates some new piece of data given what it has already generated. By sampling the posterior predictive one coin flip at a time, we can therefore *sequentially* generate the full sequence.

We start by sampling the first flip s_1, which is equally likely *a priori* to be heads or tails. This outcome informs the next coin flip, drawn according to the posterior predictive $P(s_2|s_1)$. This repeats until the final coin is predicted using all preceding results using $P(s_n|s_{n-1}, ..., s_1)$. By definition of the posterior predictive, the overall probability of generating any given sequence of observations must then equal the Bayesian evidence as

$$\begin{aligned} P(\boldsymbol{s}) &= P(s_n, s_{n-1}, ..., s_1) \\ &= P(s_n|s_{n-1}, ..., s_1)...P(s_2|s_1)P(s_1). \end{aligned} \tag{A.67}$$

From the logarithm of this equation, the description length of the data is the sum over the log-posterior-predictives at each step:

$$H(\boldsymbol{s}) = H(s_n|s_{n-1}, ..., s_1) + ... + H(s_2|s_1) + H(s_1). \tag{A.68}$$

This relationship holds regardless of the order in which the coin flips are considered. Therefore, the normalized description length is also equal to a suitably defined average

$$\frac{1}{n}H(\boldsymbol{s}) = \langle H(s_i|\boldsymbol{s}^{\text{train}})\rangle_{i,\boldsymbol{s}^{\text{train}}} \tag{A.69}$$

over all possible subsets of training data and choices of single withheld test point s_i [51].

We can thus use the Bayesian evidence not only as an information theoretic measure for model selection, but also as an indicator of overall predictive power. However, in keeping with much of the machine learning literature we will often report cross-validation results using the log-likelihood Eq. (A.64) and log-posterior-predictive Eq. (A.66) in this thesis.

Beyond prediction, we would often like to assess the quality of the inferred parameters directly. If we know from an artificial or empirical context that a parameter truly has a certain value, how does our inferred value compare? One way to establish such a "true" parameter value is in a *synthetic* test where we first draw a true value of the parameter p^{true} from the prior $P(p)$. We then sample an artificial data set $\boldsymbol{s}$ from the model likelihood $P(\boldsymbol{s}|p^{\text{true}})$. Based solely on the resulting data $\boldsymbol{s}$, we then infer the parameter p and compare it to the underlying p^{true}.

In this Bayesian setting, the posterior $P(p|\boldsymbol{s})$ is by definition precisely the distribution of the parameters p that could have resulted in the observation $\boldsymbol{s}$. Thus, the full posterior distribution gives a complete and optimal description of the truth. Compared to this benchmark, synthetic tests provide valuable test cases to understand deviations in the inferences. For example, we can examine how inferences differ when models are misspecified and do not align with the actual generative process. Understanding this robustness is crucial when applying models to real data, where they very likely do not match the real generative process.

Even when we consider the posterior of the true model, we may observe how point estimate summaries differ from the true value. Depending on how we quantify the distance between the inference $\hat{p}$ and the truth p^{true}, different point estimates may be appropriate. If we define success as only when we get the parameter exactly right (using a "one-hot" metric), we should report the MAP estimate since it maximizes this posterior probability. However, if we aim to minimize the squared error (ℓ_2 metric) of our inference, we should report the expected a posteriori (EAP) value, which provides the least squares estimate over the posterior. Thus even in the idealized scenario where the data is generated by model, our choice of metric over the parameters influences how we should summarize the inference, either with the mode or the mean of the posterior.

While we can optimize our point estimates accordingly, the posterior distribution can often be highly dispersed or even multimodal. This means that, given the data, multiple parameter values may fit equally well. The true parameter could reside at any of these peaks, meaning that no single point estimate can reliably be close to the truth. Many inference problems undergo a transition between a noisy regime where it is not possible to consistently identify the generating parameters to a data-rich regime where it becomes feasible. Section 1.2.3 discusses such an example in the context of finding group structures in networks, which corresponds to the phase transition of the Ising model at its critical temperature.

In this thesis, we will employ synthetic tests, cross-validation, and parameter metrics to better understand the performance of our network models. Applying these validation frameworks to networks presents unique challenges. For instance, when examining the group structure of a network, we need to evaluate the quality of group identity parameters. Unlike the real probability p of coin flips, there is no inherent notion of "distance" or "mean" among group labelings, which are categorical variables, to facilitate comparison.

In Chapter 3 we discuss information-theoretic measures to assess the similarity between two such clusterings of the same set of objects. We then apply this measure in synthetic tests to observe the relative performance of commonly used algorithms to recover the ground truth groups used to generate the network. In this picture we also observe regimes or types of group structure where all algorithms struggle to recover the truth.

The projects considered in this work involve ideas borrowed from the disciplines discussed in all of these Appendices, often in ways that do not cleanly separate into any single category. In Figure A.2 we have illustrated schematics of these applications across the thesis.

A.5 Equivalence of SBM inference and the Ising model

In this appendix we discuss the relationship between the Ising model, mentioned in Appendix A.2, and the stochastic block model of Section 1.2.3. More details of this correspondence may be found in the review [92].

We first consider a simpler form of the SBM. Suppose that we infer a division of the network into at most two groups. If we interpret this inference problem physically, as in Eq. (A.38), the posterior distribution of possible group structures is equivalent to the thermal ensemble of the Ising model.

To make this correspondence clear, we write the group partition as $\boldsymbol{s}$, where the entry $s_i = 1$ indicates that node i belongs to one group, while $s_i = -1$ indicates it belongs to the other. In this notation the model posterior is

$$\begin{aligned} P(\boldsymbol{s}|\boldsymbol{A}) &\propto \prod_{i<j} \omega_{b_i b_j}^{A_{ij}} \\ &\propto \prod_{(i,j)\in E} \left(\frac{p_{\text{in}}}{p_{\text{out}}}\right)^{\delta_{b_i b_j}} \end{aligned} \tag{A.70}$$

where we have set all in- and out-group weights to fixed probabilities, as in the planted partition model Eq. (2.31), and used a uniform prior over groups $P(\boldsymbol{s})$. If we compare this to the Hamiltonian of the Ising model at unit coupling $J = 1$ on the same graph,

$$H(\boldsymbol{s}) = -\sum_{(i,j)\in E} \sigma_i \sigma_j, \tag{A.71}$$

we have the equivalence

$$P(\boldsymbol{s}|\boldsymbol{A}) \propto e^{-\beta H(\boldsymbol{s})}, \tag{A.72}$$

for inverse temperature

$$\beta = \frac{1}{T} = \log\left(\frac{p_{\text{in}}}{p_{\text{out}}}\right). \tag{A.73}$$

In other words, given a network $\boldsymbol{A}$, the likely group assignments of the network for an in-group preference ratio $p_{\text{in}}/p_{\text{out}}$ are given by the Ising model configurations at the corresponding temperature.

This has profound implications for the stochastic block model. One of the key properties of the ferromagnetic Ising model is its phase transition. When the temperature is low, the model is near its ground state where all of the spins are aligned since it is energetically favorable. When the temperature is high enough, however, entropy takes over and these long range correlations dissipate. Between these two regimes, the Ising model undergoes a phase transition at a critical temperature where the correlation length diverges. The particular value of this critical temperature depends on the topology of the graph. Samples from these regimes are plotted in Figure A.11a, demonstrating the sudden breakdown of the ordered state.

Analogously, inference with the stochastic block model has two regimes shown in Figure A.11b in the constant degree limit. In the "easy" regime, when $p_{\text{out}} \ll p_{\text{in}}$, groups are well-separated and easy to identify. If we simulate an Ising model on the same network at this low temperature, the resulting configurations will be near the ground state and the true planted groups. In contrast when $p_{\text{out}} = p_{\text{in}}$ it is completely impossible to recover the groups since they have no bearing on the network structure. The corresponding Ising model has infinite temperature $T = 1/\log(p_{\text{in}}/p_{\text{out}})$ and every possible partition is equally likely. The phase transition of the Ising model divides these regimes by a *detectability threshold* in the ratio $p_{\text{in}}/p_{\text{out}}$. When the signal is above this threshold we can reliably recover the groups while below the threshold we cannot.

The precise meaning of this observation and limit can be formalized, as reviewed in [6]. For more complex SBMs with more groups and non-trivial priors on $\boldsymbol{b}$, we can consider a Potts Model with external fields. Across these settings similar detectability thresholds emerge. In our analysis on real networks in Section 3.2.3, we observe that this barrier indeed limits many community detection algorithms.

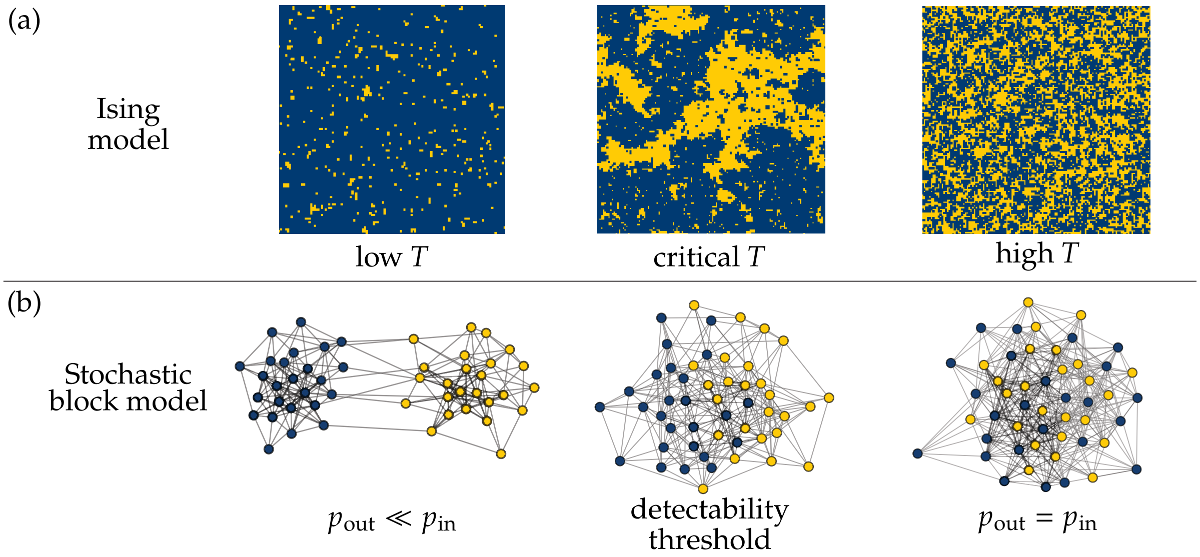


Figure A.11: The analogous phase transitions of the Ising model and stochastic block model. (a) The Ising model moves from an ordered phase at low temperature T to a disordered, noisy phase at high temperature. These phases are separated by a critical temperature at which the correlation length diverges. Samples from the model on a square lattice are shown. (b) Inference in the stochastic block model is easy when the probability of connection between groups p_{out} is much smaller than within groups p_{in}. If these probabilities are the same inference is impossible. The two regimes are separated by a detectability threshold that corresponds to the phase transition of the Ising model on that network.

Appendix B

Supplementary Material for Chapter 2

B.6 Stochastic block model Markov Chain Monte Carlo

In this appendix we describe the Monte Carlo algorithm used in order to sample from the posterior distribution of the stochastic block models described in Chapter 2. In order to sample from the posterior distribution of these models, which can be generally written as a combination of a group partition $\boldsymbol{b}$ and continuous parameters.

For generality, we consider a model with a labeling parameter $\boldsymbol{b}$, a single real parameter x and likelihood $P(\boldsymbol{A}|\boldsymbol{b}, x)$. We will consider the specific prior

$$P(\boldsymbol{b}) = \frac{\prod_r n_r!}{n!}\binom{n-1}{q-1}^{-1}\frac{1}{n} \tag{B.1}$$

and arbitrary prior $P(x)$. We would then like to sample from the posterior distribution

$$P(\boldsymbol{b}, x|\boldsymbol{A}) \propto P(\boldsymbol{A}|\boldsymbol{b}, x)P(\boldsymbol{b})P(x). \tag{B.2}$$

Our strategy for accomplishing this will be to directly sample from the prior distribution over the parameters and then use a Metropolis-Hastings accept/reject step to account for the influence of the likelihood. That is, if we have a Markov chain $P(\boldsymbol{b}, x \longrightarrow \boldsymbol{b}', x')$ that converges to the prior distribution $P(\boldsymbol{b})P(x)$, if we accept the proposed $(\boldsymbol{b}', x')$ according to the ratio

$$P_{\text{accept}}(\boldsymbol{b}, x \longrightarrow \boldsymbol{b}', x') = \min\left(1, \frac{P(\boldsymbol{A}|\boldsymbol{b}', x')}{P(\boldsymbol{A}|\boldsymbol{b}, x)}\right) \tag{B.3}$$

we are left with a Markov chain that converges to the posterior $P(\boldsymbol{b}, x|\boldsymbol{A})$. The mixing time of this chain, which determines its efficiency, depends on how often these proposals are accepted. Since the likelihood $P(\boldsymbol{A}|\boldsymbol{b}, x)$ is often a rapidly varying function of the parameters, we will therefore only propose small changes in the parameters since values far away from the current value are likely to be rejected. Furthermore, the change in the likelihood after a local move can often be computed more efficiently than a completely new set of parameters. Although effective Markov chains have been proposed that instead aim for larger global moves in order to improve ergodicity of the chain [108], we use local chains for simplicity in this thesis.

Since the groups $\boldsymbol{b}$ and parameter x are independent in the prior distribution we can sample them separately. To sample x, we first compute its cumulative distribution $F(x)$,

$$F(x) = \int_{-\infty}^{x} P(x')dx'. \tag{B.4}$$

Assuming $P(x)$ is supported everywhere, $F : \mathbb{R} \to [0,1]$ is a monotonically increasing function and can be inverted as $F^{-1} : [0,1] \to \mathbb{R}$. To generate a sample from $P(x)$ we can then first sample uniformly from the unit interval $u \in [0,1]$ to obtain the sample $x = F^{-1}(u)$.

Although this process directly generates samples from the distribution, these independent samples are likely to large differences between them that will slow down the accept/reject step on the likelihood. To generate smaller moves, we will adapt our sampling procedure into a Markov chain that converges to $P(\boldsymbol{x})$. We first define a Markov chain that generates uniform samples $u \in [0,1]$ on the unit interval. At each step we add Gaussian noise of some width $\sigma > 0$ as

$$P(u \to u') = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\frac{(u'-u)^2}{2\sigma^2}} \tag{B.5}$$

then implement reflective boundary conditions on the interval. At each step we can then simply transform this unit interval samples into samples of x that are highly correlated as $x = F^{-1}(u)$. The choice of step size σ here is arbitrary, and is a compromise between not being too large to avoid rejections, while not being too small to efficiently explore the space.

For the partition $\boldsymbol{b}$, although we could again sample directly from the distribution Eq. (B.1) fairly easily, we will construct a Markov chain of local moves over the space of partitions that converges to the appropriate distribution. This is a fairly complicated task, as these proposals must be able to change not only the assignments of the nodes to each group but also change the number of groups q. Here we summarize the Markov chain used by [116] for this purpose.

The scheme consists of two types of Monte Carlo moves, simplified as:

- **Type 1:** If possible, choose a random pair of distinct groups r, s then randomly choose a node from group r to move into group s. If this act empties group r, remove that group and relabel the nodes appropriately.

- **Type 2:** Choose a random group s and then a random node from that group to move a new group on its own. If this empties group s relabel appropriately.

If we then propose a Type 1 move with probability $1 - 1/n$ and a Type 2 move with probability $1/n$, the resulting Markov chain will converge to the distribution $P(\boldsymbol{b})$.

We can now make proposals to change the partition $\boldsymbol{b}$ and the continuous parameter x respecting their prior distributions. In practice, we propose changes to the partition $\boldsymbol{b}$ a factor n times more often than to each continuous parameter x since the partition is effectively a separate parameter on each node. After attenuating the samples by the likelihood using the accept/reject step we are then left with a Markov chain that will converge to the appropriate posterior distribution.

To interpret the results from this chain as representative independent samples from the posterior distribution, we measure the auto-correlations between the samples to estimate the mixing time, then only record samples with a gap between them on that order. The added time needed to wait between samples in large systems is the main computational impediment to applying these types of Bayesian techniques to very large systems, although we are able to get fairly consistent results on circa 2025 consumer hardware for networks with tens of thousands of nodes in this work.

B.7 Consensus clustering

In this appendix we describe how posterior distributions over possible group partitions are summarized into a single consensus clustering. Our goal in this task is to find a single partition that best reflects the typical behavior of the full posterior distribution.

For a given partition $\boldsymbol{b}$ we can define an $n \times n$ *coincidence matrix* $C[\boldsymbol{b}]$ whose entries reflect whether two nodes belong to the same group:

$$C[\boldsymbol{b}]_{ij} = \delta_{b_i b_j}. \tag{B.6}$$

Given T sampled partitions $\boldsymbol{b}_t$ for $t = 1, ..., T$ we can then compute the *aggregate* coincidence matrix as the average

$$\bar{\boldsymbol{C}} = \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{C}[\boldsymbol{b}_t]. \tag{B.7}$$

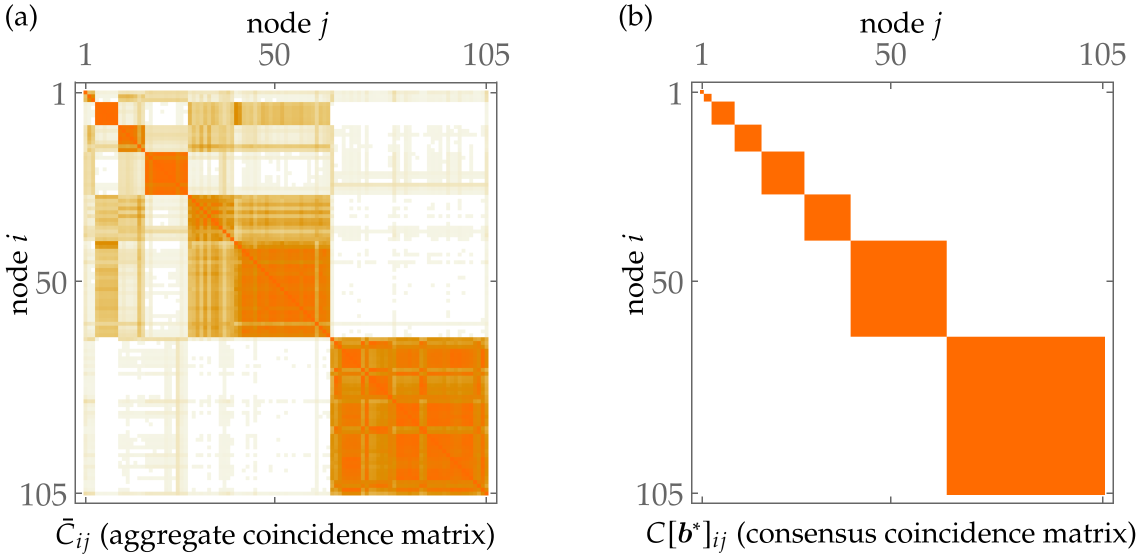


Figure B.1: Consensus clustering found by the general degree-corrected SBM on the network of $n = 105$ books shown in Figure 2.2c. (a) Entries of the aggregate coincidence matrix $\bar{\boldsymbol{C}}$ taken from the posterior distribution. (b) The coincidence matrix of the found consensus clustering $\boldsymbol{b}^*$, defined by closeness to the aggregate coincidence.

Entry $\bar{C}_{ij}$ of this aggregate matrix then gives the fraction of samples where nodes i and j belong to the same group.

In principle, we would like for our consensus clustering to reflect this posterior distribution. If two nodes are typically in the same group across the posterior distribution, they should be placed in the same group in the consensus clustering. Inevitably restricting to a single consensus clustering can not capture the full range of possible aggregate coincidence matrices, for example that two nodes belong to the same group only a fraction of the time, we can only hope that the consensus is an approximation of the truth.

We then look to find the partition $\boldsymbol{b}^*$ whose coincidence matrix is closest to the aggregate coincidence matrix in an L_2 (or Frobenius) sense,

$$\boldsymbol{b}^* = \mathrm{argmin}_{\boldsymbol{b}} \left\| \boldsymbol{C}[\boldsymbol{b}] - \bar{\boldsymbol{C}} \right\|_2. \tag{B.8}$$

In practice, this distance is optimized over the samples $\boldsymbol{b}_t$ taken to inform the coincidence matrix itself. The overall complexity of these operations is therefore $O(Tq^2)$, the cost to construct the aggregate coincidence matrix. Figure B.1 shows the results of these coincidence matrices to summarize an SBM posterior distribution.

B.8 General posterior-predictive

In this appendix we write out the full posterior-predictive distribution of the stochastic block model incorporating both the general degree correction of Section 2.1 and the assortative generalizations of Section 2.2. The posterior-predictive distributions of the particular models that this form generalizes can be found by plugging in the corresponding parameter values given in Tables 1.2 and 2.3.

Suppose that we would like to compute the posterior predictive distribution over possible testing data sets $\boldsymbol{A}^{\text{test}}$ given a training data set $\boldsymbol{A}^{\text{train}}$ and a factor $f \geq 0$ that accounts for the relative overall density of the two data sets. For example in the 80/20 cross-validation splits into training and testing data sets used in this thesis, this factor is $f = 0.25$. This factor relates how we extrapolate the weight matrix of the training data to the testing data as $\boldsymbol{\omega}^{\text{test}} = f\boldsymbol{\omega}^{\text{train}}$.

Integrating and summing over all latent parameters the posterior predictive can be written as

$$\begin{aligned}
&P(\boldsymbol{A}^{\text{test}}|\boldsymbol{A}^{\text{train}}, f)\\
&\quad= \sum_{\boldsymbol{b}} \int P(\boldsymbol{A}^{\text{test}}|\boldsymbol{\theta}, f\boldsymbol{\omega}, \boldsymbol{b}, \alpha, \rho_{\text{in}}, \rho_{\text{out}}, \lambda_{\text{in}}, \lambda_{\text{out}})\\
&\qquad\qquad \times P(\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{b}, \alpha, \rho_{\text{in}}, \rho_{\text{out}}, \lambda_{\text{in}}, \lambda_{\text{out}}|\boldsymbol{A}^{\text{train}}) d\boldsymbol{\theta} d\boldsymbol{\omega} d\alpha d\rho_{\text{in}} d\rho_{\text{out}} d\lambda_{\text{in}} d\lambda_{\text{out}}\\
&\quad= \sum_{\boldsymbol{b}} \int P(\boldsymbol{A}^{\text{test}}|\boldsymbol{A}^{\text{train}}, \boldsymbol{b}, \alpha, \rho_{\text{in}}, \rho_{\text{out}}, \lambda_{\text{in}}, \lambda_{\text{out}}, f) \qquad\qquad (\text{B.9})\\
&\qquad\qquad \times P(\boldsymbol{b}, \alpha, \rho_{\text{in}}, \rho_{\text{out}}, \lambda_{\text{in}}, \lambda_{\text{out}}|\boldsymbol{A}^{\text{train}}) d\alpha d\gamma d\rho_{\text{in}} d\rho_{\text{out}}.
\end{aligned}$$

In the last line of this equation we have integrated over the node weights $\boldsymbol{\theta}$ and the weight matrix $\boldsymbol{\omega}$ to obtain

$$\begin{aligned}
&P(\boldsymbol{A}^{\text{test}}|\boldsymbol{A}^{\text{train}}, \boldsymbol{b}, \alpha, \rho_{\text{in}}, \rho_{\text{out}}, \lambda_{\text{in}}, \lambda_{\text{out}}, f) &(\text{B.10})\\
&\quad= P(\boldsymbol{A}^{\text{test}}|\boldsymbol{k}^{\text{test}}, \boldsymbol{M}^{\text{test}}, \boldsymbol{b}) &(\text{B.11})\\
&\qquad \times P(\boldsymbol{M}^{\text{test}}|\boldsymbol{M}^{\text{train}}, \boldsymbol{n}, \rho_{\text{in}}, \rho_{\text{out}}, \lambda_{\text{in}}, \lambda_{\text{out}}, f) &(\text{B.12})\\
&\qquad \times P(\boldsymbol{k}^{\text{test}}|\boldsymbol{k}^{\text{train}}, \boldsymbol{m}^{\text{test}}, \boldsymbol{m}^{\text{train}}, \boldsymbol{n}, \alpha). &(\text{B.13})
\end{aligned}$$

This expression contains a stub-matching likelihood of the network

$$P(\boldsymbol{A}^{\text{test}}|\boldsymbol{k}^{\text{test}}, \boldsymbol{M}^{\text{test}}, \boldsymbol{b}) = \frac{\prod_i k_i^{\text{test}}! \prod_{r<s} M_{rs}^{\text{test}}! \prod_r M_{rr}^{\text{test}}!!}{\prod_{i<j} A_{ij}^{\text{test}}! \prod_i A_{ii}^{\text{test}}!! \prod_r m_r^{\text{test}}!} \qquad (\text{B.14})$$

and negative binomial distributions of the edge count matrix entries

$$\begin{aligned}
&P(\boldsymbol{M}^{\text{test}}|\boldsymbol{M}^{\text{train}}, \boldsymbol{n}, \rho_{\text{in}}, \rho_{\text{out}}, \lambda_{\text{in}}, \lambda_{\text{out}}, f) \qquad \text{(B.15)}\\
&\quad = \int P(\boldsymbol{M}^{\text{test}}|f\boldsymbol{\omega}, \boldsymbol{n}, \rho_{\text{in}}, \rho_{\text{out}}, \lambda_{\text{in}}, \lambda_{\text{out}})P(\boldsymbol{\omega}|\boldsymbol{M}^{\text{train}}, \boldsymbol{n}, \rho_{\text{in}}, \rho_{\text{out}}, \lambda_{\text{in}}, \lambda_{\text{out}})d\boldsymbol{\omega}\\
&\quad = \prod_{r<s} \binom{M_{rs}^{\text{test}} + M_{rs}^{\text{train}} + \lambda_{\text{out}} - 1}{M_{rs}^{\text{train}} + \lambda_{\text{out}} - 1} \frac{(f\rho_{\text{out}}n_r n_s)^{M_{rs}^{\text{test}}}(\rho_{\text{out}}n_r n_s + \lambda_{\text{out}})^{M_{rs}^{\text{train}}+\lambda_{\text{out}}}}{((1+f)\rho_{\text{out}}n_r n_s + \lambda_{\text{out}})^{M_{rs}^{\text{test}}+M_{rs}^{\text{train}}+\lambda_{\text{out}}}}\\
&\quad \times \prod_r \binom{M_{rr}^{\text{test}}/2 + M_{rr}^{\text{train}}/2 + \lambda_{\text{in}} - 1}{M_{rr}^{\text{train}}/2 + \lambda_{\text{in}} - 1} \frac{(f\rho_{\text{in}}n_r^2/2)^{M_{rr}^{\text{test}}/2}(\rho_{\text{in}}n_r^2/2 + \lambda_{\text{in}})^{M_{rr}^{\text{train}}/2+\lambda_{\text{in}}}}{((1+f)\rho_{\text{in}}n_r^2/2 + \lambda_{\text{in}})^{M_{rr}^{\text{test}}/2+M_{rr}^{\text{train}}/2+\lambda_{\text{in}}}}
\end{aligned}$$

with expectations

$$\mathbf{E}M_{rs}^{\text{test}} = \begin{cases} f\rho_{\text{out}}n_r n_s \frac{M_{rs}^{\text{train}}+\lambda_{\text{out}}}{\rho_{\text{out}}n_r n_s+\lambda_{\text{out}}} & r \neq s \\ f\rho_{\text{in}}n_r^2/2\frac{M_{rr}^{\text{train}}/2+\lambda_{\text{in}}}{\rho_{\text{in}}n_r^2/2+\lambda_{\text{in}}} & r = s \end{cases}. \qquad \text{(B.16)}$$

We note here that the factor f just leads to an overall scaling of the edge count matrix. Also, for the homogeneity parameters $\lambda_{\text{in}} = \lambda_{\text{out}} = 1$, as found in the simple assortative SBM, these negative binomial distributions simplify to a geometric distributions of the same means.

The predicted degree distribution is a Dirichlet-multinomial distribution

$$P(\boldsymbol{k}^{\text{test}}|\boldsymbol{k}^{\text{train}}, \boldsymbol{m}^{\text{test}}, \boldsymbol{m}^{\text{train}}, \boldsymbol{n}, \alpha) = \qquad \text{(B.17)}$$

$$\prod_r \binom{m_r^{\text{test}} + m_r^{\text{train}} + n_r\alpha - 1}{m_r^{\text{train}} + n_r\alpha - 1}^{-1} \prod_{i\in r} \binom{k_i^{\text{test}} + k_i^{\text{train}} + \alpha - 1}{k_i^{\text{train}} + \alpha - 1} \qquad \text{(B.18)}$$

with expectations

$$\mathbf{E}k_i^{\text{test}} = m_{b_i}^{\text{test}} \frac{k_i^{\text{train}} + \alpha}{m_{b_i}^{\text{train}} + n_{b_i}\alpha} \qquad \text{(B.19)}$$

and Dirichlet parameters

$$\alpha_i = k_i^{\text{train}} + \alpha. \qquad \text{(B.20)}$$

To compute the final posterior-predictive, the integral Eq. (B.9) is then approximated by summing over MCMC samples of the parameters drawn from the training data posterior $P(\boldsymbol{b}, \alpha, \rho_{\text{in}}, \rho_{\text{out}}, \lambda_{\text{in}}, \lambda_{\text{out}}|\boldsymbol{A}^{\text{train}})$.

B.9 Hierarchical degree prior

The microcanonical formulation of the stochastic block model often uses a hierarchical prior [107] on the degree distribution within each group in a manner that does not cleanly fit into our general degree-correction. In this appendix, however, we demonstrate a choice of in-group degree homogeneity α_{in} that approximates this hierarchical prior, and show that in most of the data sets we consider this type of degree-correction is therefore excluded.

The hierarchical prior of the degrees $\boldsymbol{k}_r$ within a group r of n_r nodes of total degree m_r is defined as

$$P(\boldsymbol{k}_r|m_r, n_r) = P(\boldsymbol{k}_r|\boldsymbol{\eta})P(\boldsymbol{\eta}|m_r, n_r) \tag{B.21}$$

where the entries of $\boldsymbol{\eta}$ contain the number of vertices in group r of each degree. This $\boldsymbol{\eta}$ therefore represents a *restricted partition* of m_r into at most n_r integers. In this prior the vector $\boldsymbol{\eta}$ is distributed uniformly among the $q(m_r, n_r)$ such restricted partitions as

$$P(\boldsymbol{\eta}|m_r, n_r) = q(m_r, n_r)^{-1}. \tag{B.22}$$

Finding $q(m_r, n_r)$ is a combinatorially difficult problem, although well-behaved approximations of it exist. Given the node counts $\boldsymbol{\eta}$, the degrees are then multinomially distributed among the possible ways that the chosen degrees can be assigned to the n_r nodes as

$$P(\boldsymbol{k}_r|\boldsymbol{\eta}) = \frac{\prod_{k=0}^{m_r} \eta_k!}{n_r!}. \tag{B.23}$$

The non-analyticity of this distribution means that no choice of α_{in} will make it equal to our analytic prior Eq. (2.20), so we will need to resort to an approximation to compare the performance of the priors. Our strategy is to choose α_{in} so that the two priors exhibit the same behavior in the limit of large size and total degree $n_r, m_r \to \infty$. As shown in [107], in the $n_r \to \infty$ limit the average number of nodes η_k with degree k approaches

$$\langle \eta_k \rangle \approx \frac{1}{\exp(k\pi/\sqrt{6m_r}) - 1}. \tag{B.24}$$

To repeat this calculation of the marginal distribution for our prior

$$\boldsymbol{k}_r \sim \text{DM}(\overbrace{\alpha_{\text{in}}, \ldots, \alpha_{\text{in}}}^{n_r}) \tag{B.25}$$

we first make use of the aggregation property of the Dirichlet-multinomial distribution to show the distribution of the degree of a single node k (and its remainder) is

$$(k, m_r - k) \sim \mathrm{DM}(\alpha_{\text{in}}, (n_r - 1)\alpha_{\text{in}}). \tag{B.26}$$

The marginal distribution of the degree k is therefore

$$P(k) = \frac{\Gamma(m_r + 1)\Gamma(n_r\alpha_{\text{in}})}{\Gamma(m_r + n_r\alpha_{\text{in}})} \frac{\Gamma(k + \alpha_{\text{in}})}{\Gamma(\alpha_{\text{in}})\Gamma(k + 1)} \frac{\Gamma(m_r - k + (n_r - 1)\alpha)}{\Gamma((n_r - 1)\alpha)\Gamma(m_r - k + 1)}. \tag{B.27}$$

If we consider this distribution in the limit of infinite group size $n_r \to \infty$, the concentration parameter must scale as $\alpha_{\text{in}} = b\frac{\sqrt{m_r}}{n_r}$ for some $b > 0$ to give a fixed limiting count of nodes of a given degree,

$$\langle \eta_k \rangle = n_r P(k) \to \frac{\Gamma(m_r + 1)\Gamma(m_r - k + b\sqrt{m_r})b\sqrt{m_r}}{\Gamma(m_r - k + 1)\Gamma(m_r + b\sqrt{m_r})k}. \tag{B.28}$$

Although this $n_r \to \infty$ form is not identical to the hierarchical Eq. (B.24), in the large total degree limit $m_r \to \infty$ distribution it becomes

$$\langle n_k \rangle \approx b\frac{\sqrt{m_r}}{k}. \tag{B.29}$$

Matching this to the behavior of the hierarchical prior in the same $m_r \to \infty$ limit, we find that for the choice

$$\alpha_h = \frac{\sqrt{6m_r}}{\pi n_r}. \tag{B.30}$$

the hierarchical and general degree-corrected prior share the same limiting behavior. Figure B.2 plots example sampled degree distributions from the hierarchical and general priors for parameter values along this limit, and we can indeed observe similar distributions for this choice of parameter.

In Table B.1 we compare these approximate values of the in-group degree homogeneity t the values inferred by the general degree-corrected model. To obtain a single parameter value for he full network, rather than each individual group, we further assume the nodes are split into q equally sized groups so that

$$\alpha_h = \frac{6pq}{\pi}. \tag{B.31}$$

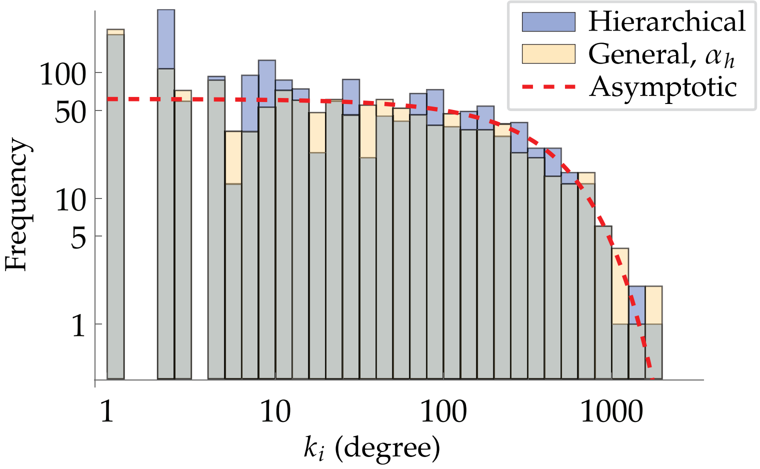


Figure B.2: Sampled degree distributions over $n_r = 10000$ nodes of average degree $m_r/n_r = 10$ from the hierarchical prior and general prior for the concentration parameter α_h. Both distributions are compared against the asymptotic distribution Eq. (B.24).

Across the data sets we consider, most data sets exclude this special case, suggesting that the general degree correction is favored over the hierarchical prior. After transforming to an effective levels of in-group degree inequality G_h, we observe that in most cases the hierarchical prior assumes a much more unequal distribution of degrees within groups than is present in the networks.

B.10 Resolution limit

In this appendix we derive the resolution limit for the usual SBM and show how it is circumvented by the assortative SBM of Section 2.2.

The illustration of the resolution limit used in Figure 2.5 is a special case of a larger class of examples used to show the resolution limit. Suppose that a network is divided into q entirely disconnected groups that serve as the true planted partition. We consider a symmetric example where the nodes and edges are equally split among the groups, so that group r has $n_r = n/q$ nodes and total degree $m_r = 2m/q$. The only non-zero entries of the edge count matrix $\boldsymbol{M}$ are then the diagonal $M_{rr} = 2m/q$. The network thus has an in-group density $\rho_{\text{in}} = q\frac{2m}{n^2}$, out-group density $\rho_{\text{out}} = 0$ and overall density $\rho = \frac{2m}{n^2}$.

To show the resolution limit, we would like to compare the description length of this true partition into q groups with a coarsening into $q^* < q$ groups where sets of $\frac{q}{q^*}$ original groups are merged. In this new partition the edge count matrix is still diagonal, with entries $M_{rr} = 2m/q^*$. For a model to pass this check and identify the original partition,

Data set	G_{in}	G_h	α_h
Football	(0.00, 0.03)	0.40	1.77
Power grid	(0.00, 0.02)	0.99	0.01
Karate	(0.00, 0.16)	0.55	0.76
Books	(0.00, 0.29)	0.49	1.07
Food web	(0.07, 0.11)	0.16	12.09
Friends	(0.14, 0.21)	0.45	1.29
Dolphins	(0.12, 0.26)	0.59	0.63
Neurons	(0.23, 0.28)	0.43	1.49
Proteins	(0.27, 0.29)	0.94	0.05
Coauthors	(0.28, 0.31)	0.95	0.04
Words	(0.26, 0.35)	0.55	0.78
Internet	(0.33, 0.35)	0.97	0.02
E-mail	(0.41, 0.43)	0.69	0.37
Blogs	(0.59, 0.61)	0.60	0.60

Table B.1: Data sets of Table 2.2 along with an inter-quartile range of the inferred Gini coefficient of in-group degree inequality G_{in}. These are compared against the values of the index G_h and the concentration parameter α_h that would asymptotically recover the behavior of the hierarchical degree prior as in Eq. (B.31). Most cases exclude this special case in favor of the general degree-corrected model.

the description length of the coarsening must be larger than that of the original partition for all $q^* < q$. Otherwise the coarse partition would be preferred over the truth. The usual SBM fails this check in the example of Figure 2.5 and prefers $q^* = 4$ over $q = 8$.

To show when this occurs we analyze the description length of the SBM. To get analytic control over the description length we consider it in a constant degree limit of large network size $n, m \to \infty$. We also assume that the number of communities in the network grows sublinearly as $q \sim o(n)$ so that each community grows $\frac{m}{q} \to \infty$. Although this excludes the limit where community size is constant, we will find the resolution limit of the model well before this point.

This is a sparse graph limit, even within the groups as $\rho_{\text{in}} = q\frac{2m}{n^2} = \frac{qc^2}{2m} \to 0$. We can therefore assume that the graph is simple and expand the encoding cost of the network for small q (or large $\frac{m}{q}$) as

$$\begin{aligned} H(\boldsymbol{A}|\boldsymbol{M},\boldsymbol{b}) &= -\log\left[\prod_r \frac{M_{rr}!!}{n_r^{m_r}}\right] \\ &= 2m\log\left(\frac{n}{q}\right) - q\log\left[(2m/q)!!\right] \\ &\approx m\left[1+\log\left(\frac{n^2}{2mq}\right)\right] + O(m/q). \end{aligned} \tag{B.32}$$

The edge count matrix encoding dominates the total information cost for large q, so we instead expand it in small m/q as

$$\begin{aligned} H(\boldsymbol{M}|\boldsymbol{n},q) &= \frac{q(q-1)}{2}\log\left(2m/q^2+1\right) + q\left[(m/q+1)\log(m/q^2+1) - m/q\log(m/q^2)\right] \\ &= m\left[1-\log\left(\frac{m}{q^2}\right)\right] + O(m/q). \end{aligned} \tag{B.33}$$

To complete this approximation, we add a non-perturbative correction to match the small q behavior $H(\boldsymbol{M}|\boldsymbol{n}, q=0) = 0$ as

$$H(\boldsymbol{M}|\boldsymbol{n},q) = m\left[1-\log\left(\frac{m}{q^2+m/e}\right)\right] + O(m/q). \tag{B.34}$$

Taken together, these description lengths approximate the behavior of the description length in this limit. Figure B.3a shows an example of this encoding cost for networks with $n = 5000$ nodes, $m = 10000$ edges, and varying numbers of groups q. We can observe that both the true description length and the approximation have a characteristic "U" shape.

To find the minimum of this curve, we can differentiate the approximate description length

$$\partial_q H(\boldsymbol{A}|\boldsymbol{b},\rho) \approx \frac{m}{q}\frac{eq^2-m}{eq^2+m} \tag{B.35}$$

and identify the critical number of communities $q_c = \sqrt{m/e}$.

The critical number of communities $q_c = \sqrt{m/e}$ therefore splits the number of communities into two regimes. When $q < q_c$, no coarsening $q^* < q$ will return a smaller description length than the true partition. For $q > q_c$, however, the description length will instead be minimized at $q^* = q_c < q$, and the model will not infer the true number of

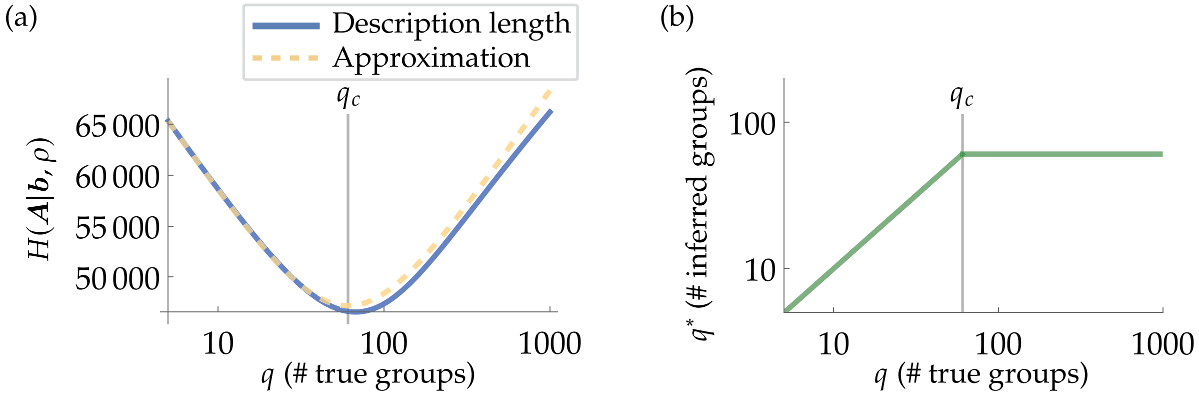


Figure B.3: (a) Plot of the description length of a fully assortative network as the number of true groups q varies. (b) True number of planted groups q and the inferred number of groups q^*. Beyond the resolution limit when $q > q_c$ the usual SBM cannot infer more than the critical number of groups.

groups. This effect is illustrated in Figure B.3b.

We can compare this to the behavior of the simple assortative SBM in the same limit. This model shares the description length of the network $H(\boldsymbol{A}|\boldsymbol{M},\boldsymbol{b})$, although the information cost of the edge count matrix is now

$$H(\boldsymbol{M}|\boldsymbol{n},\rho_{\text{in}},\rho_{\text{out}}) = q\left[(m/q+1)\log(m/q+1) - m/q\log(m/q)\right] \tag{B.36}$$

$$= m\left[1-\log\left(\frac{m}{q+m/e}\right)\right] + O(m/q). \tag{B.37}$$

The derivative of the overall description length in the assortative model is then approximately

$$\partial_q H(\boldsymbol{A}|\boldsymbol{b},\rho_{\text{in}},\rho_{\text{out}}) \approx -\frac{m^2}{eq^2+mq}, \tag{B.38}$$

which is always negative, indicating that in this regime there is no smaller number of communities $q^* < q$ preferred over the true partition, and the resolution limit is alleviated.

There are still some limitations of the number of communities we can infer in the assortative model. If we include the influence of the prior $P(\boldsymbol{b})$ on the group structure, the MAP estimate will have at most $O(m/\log(m))$ communities, far more than the $O(\sqrt{m})$ identified by the usual model. Additionally, if the communities are not completely assortative, and so the model must deal with a noisy signal, the assortative models are still limited by the detectability threshold discussed in Appendix A.5.

B.11 Synthetic tests

In our "Symmetric" synthetic tests of Section 2.2.5, we consider networks of a fixed mean degree c and fixed signal to noise ratio (SNR)

$$\mathrm{SNR} = \frac{n^2(\rho_{\mathrm{in}} - \rho_{\mathrm{out}})^2}{cq^2}. \tag{B.39}$$

This choice is made to make the challenge of reconstructing the true groups a similar difficulty over all the examples regardless of the number of groups q. This choice results in the densities

$$\rho_{\mathrm{in}} = \frac{c}{n}\left(1 + (q-1)\sqrt{\frac{\mathrm{SNR}}{c}}\right), \quad \rho_{\mathrm{out}} = \frac{c}{n}\left(1 - \sqrt{\frac{\mathrm{SNR}}{c}}\right) \tag{B.40}$$

which then inform the edge count matrix entries as

$$M_{rs} = \frac{n^2}{q^2} \times \begin{cases} \rho_{\mathrm{in}} & r = s \\ \rho_{\mathrm{out}} & r \neq s. \end{cases} \tag{B.41}$$

We note that SNR = c recovers the usual resolution limit test of completely disconnected groups as $\rho_{\mathrm{out}} = 0$. Our choice of $c = 5$ and SNR = 3 is slightly more challenging than this case, although we observe in Figure 2.7 that the usual SBM breaks down at a similar resolution limit as we would expect from the analysis in Section B.10, $q = \sqrt{m/e} \approx 30.3$ for $m = 2500$.

Admittedly, the use of the signal-to-noise ratio is slightly dubious in this context, especially as the number of groups approaches the total number of nodes. The detectability threshold SNR > 1 tells us only when recoverability is possible in the limit of an infinitely large network and a fixed number of groups [6]. Yet, this is a useful framework for our purposes.

In our tests we also consider a "Ring" structure whose groups connect only as neighbors along a ring and where there is no global assortative pattern, $\rho_{\mathrm{in}} = \rho_{\mathrm{out}}$. These conditions set the edge count matrix entries[1]

$$M_{rs} = \frac{cn}{q^2} \times \begin{cases} 1 & r = s \\ \frac{q-1}{2} & r = s \pm 1 (\mathrm{mod}\ q) \\ 0 & \text{else.} \end{cases} \tag{B.42}$$

Like the symmetric test, we then generate our synthetic networks on $n = 1000$ nodes with average degree $c = 5$ for our tests.

B.12 Comparison to modularity

In this appendix we compare the assortativity inferred by our simple assortative SBM presented in Section 2.2 to the modularity of the partitions it finds. We show that these are related, yet contrasting measures of the overall strength of group structure in a network.

As discussed in Section 1.2.3, the modularity of a network $\boldsymbol{A}$ over a partition $\boldsymbol{b}$ measures the (normalized) number of edges m_{in} within the groups above the number $\langle m_{\text{in}} \rangle_{\text{config}}$ expected in the configuration model:

$$Q(\boldsymbol{A}, \boldsymbol{b}) = \frac{m_{\text{in}} - \langle m_{\text{in}} \rangle_{\text{config}}}{m}. \tag{B.43}$$

Figure B.4 compares this modularity to the assortativity for the networks listed in Table 2.2. We see that both measures qualitatively agree on which of the networks are assortative ($Q > 0, \rho_{\text{in}} > \rho_{\text{out}}$) and disassortative ($Q < 0, \rho_{\text{in}} < \rho_{\text{out}}$) with the possible exception of the "Proteins" network. The two measures, however, vary wildly in their assessment of how assortative each network is. For instance, while the "Email" and "Dolphins" data sets share a modularity of $Q \approx 0.25$, the "Email" data set is strongly assortative at $\rho_{\text{in}}/\rho_{\text{out}} \approx 16$, an order of magnitude greater than the weak $\rho_{\text{in}}/\rho_{\text{out}} \approx 1.3$ preference found in the dolphins.

To understand the source of this discrepancy we can consider the relationship between the two measures in a simple test case. Suppose that the nodes are split into q equal groups. Of the $\frac{n^2}{2}$ unique pairs of nodes in the network, $\frac{n^2}{2q}$ are between nodes of the same group. At an in-group density of ρ_{in} we therefore expect to observe $m_{\text{in}} = \rho_{\text{in}} \frac{n^2}{2q}$ edges within the groups. If we further assume that the nodes all have the same degree, the configuration model would expect that $1/q$ of the edges like within groups as $\langle m_{\text{in}} \rangle_{\text{config}} =$

[1] For $q = 2$ the off-diagonal entries are doubled.

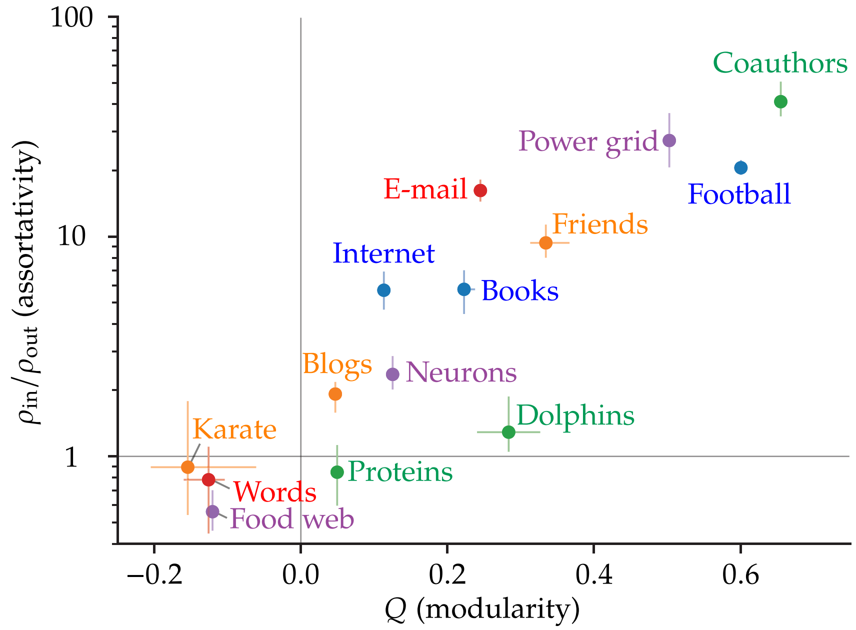


Figure B.4: Figure of inferred group assortativity $\rho_{\text{in}}/\rho_{\text{out}}$ versus the modularity Q of partitions sampled by the simple ASBM. $Q = 0$ and $\rho_{\text{in}}/\rho_{\text{out}} = 1$ are highlighted to delineate between the assortative and disassortative regimes.

m/q. Assembling these expectations yields the typical modularity

$$Q = \frac{\rho_{\text{in}} n^2/(2q) - m/q}{m} = \frac{1}{q}\left(\frac{\rho_{\text{in}}}{\rho} - 1\right). \tag{B.44}$$

This simple example demonstrates a number of the patterns found in our real networks. For one, when a network is assortative, $\rho_{\text{in}} > \rho$, the modularity is positive, and the opposite holds for disassortative networks. The modularity also scales inversely with the number of groups when the assortativity is held fixed. If individuals maintain the same level of assortative preference in their connections as the number of groups increases, fewer connections are made inside groups since the groups become smaller. To maintain the same overall ratio of edges within increasingly fragmented groups, individuals must adopt a higher in-group preference to maintain the same in-group/out-group composition of their interactions.

Returning to our earlier comparison, the "Email" network has $q = 23$ groups while the dolphins only have 4. For both networks to have a similar overall modularity, emails must be sent with a far stronger assortative preference than the dolphins exhibit since the relative sizes of the in-group and out-group vary drastically between the cases.

In summary, the assortativity directly measures the microscopic in-group preference nodes adopt whereas the modularity reflects the resulting global difference in the composition of edges within and between groups. Depending on the pattern of group affiliations, the measures thus give complementary perspectives on the nature of the group structure.

Appendix C

Supplementary Material for Chapter 3

C.13 Upper bound on the normalized mutual information

In this appendix we show that the value of the normalized mutual information measure proposed in this paper is bounded above by 1:

$$\mathrm{NMI}_{\mathrm{DM}}(c;g) = \frac{I_{\mathrm{DM}}(c;g)}{I_{\mathrm{DM}}(g;g)} \leq 1, \tag{C.1}$$

and moreover that the exact equality is achieved if and only if c and g are identical up to a permutation of their labels. These properties ensure that no candidate can receive a score higher than that of the ground truth itself and enable us to interpret a score of 1 as equality up to permutation. The ordinary non-reduced NMI, normalized as in Eq. (3.35), does not have the same properties—as shown in Fig. 3.4, there are possible labelings c that are substantially different from g but nonetheless give a conventional NMI of 1. It is possible that the standard ("flat") reduced mutual information satisfies a bound like (C.1), but this has not been proven. It is known to be violated if poor approximations of $\Omega(n^{(g)}, n^{(c)})$ are used, so any proof would require an exact expression for $\Omega(n^{(g)}, n^{(c)})$ or a sufficiently good estimate. It is unclear whether current estimates are good enough, although we are not aware of any violations of the relevant inequality when the estimate of Eq. (3.14) is employed.

To prove (C.1) we express the numerator and denominator as

$$\begin{aligned} I_{\mathrm{DM}}(c;g) &= I_0(c;g) + H(n^{(g)}|n, q_g, \alpha_g) \\ &\qquad - H(n^{(gc)}|n^{(c)}, q_g, \alpha_{g|c}), \end{aligned} \tag{C.2}$$

$$\begin{aligned} I_{\mathrm{DM}}(g;g) &= I_0(g;g) + H(n^{(g)}|n, q_g, \alpha_g) \\ &\qquad - H(n^{(gg)}|n^{(g)}, q_g, \alpha_{g|g}) \\ &= H_0(g) + H(n^{(g)}|n, q_g, \alpha_g) - q_g \log q_g, \end{aligned} \tag{C.3}$$

as in Eqs. (3.31) and (3.34). Then our desired bound can be rewritten as

$$\log \frac{\prod_s n_s^{(c)}!}{\prod_{rs} n_{rs}^{(gc)}!} + H(n^{(gc)}|n^{(c)}, q_g, \alpha_{g|c}) \geq q_g \log q_g. \tag{C.4}$$

The left-hand side of this inequality decreases when the entries of the contingency table decrease. To demonstrate this we define a table $\tilde{n}^{(gc)}$ which is identical to the original table $n^{(gc)}$ except that a single entry is decreased by 1: $\tilde{n}_{rs}^{(gc)} = n_{rs}^{(gc)} - 1$. With this change the first term in Eq. (C.4) must decrease, since

$$\log \frac{\prod_s n_s^{(c)}!}{\prod_{rs} n_{rs}^{(gc)}!} - \log \frac{\prod_s \tilde{n}_s^{(c)}!}{\prod_{rs} \tilde{n}_{rs}^{(gc)}!} = \log \frac{n_s^{(c)}}{n_{rs}^{(gc)}} \geq 0. \tag{C.5}$$

Similarly, the second term in Eq. (C.4) also decreases if we make the further assumption that

$$n_{rs}^{(gc)} \geq n_s^{(c)}/q_g, \qquad n_{rs}^{(gc)} > 1. \tag{C.6}$$

Under these conditions we can bound the change in the second term by

$$\begin{aligned}
&H(n^{(gc)}|n^{(c)}, q_g, \alpha_{g|c}) - H(\tilde{n}^{(gc)}|\tilde{n}^{(c)}, q_g, \alpha_{g|c}) \\
&\quad = \log \binom{n_s^{(c)} + q_g\alpha_{g|c} - 1}{q_g\alpha_{g|c} - 1} - \log \binom{\tilde{n}_s^{(c)} + q_g\alpha_{g|c} - 1}{q_g\alpha_{g|c} - 1} \\
&\qquad - \log \binom{n_{rs}^{(gc)} + \alpha_{g|c} - 1}{\alpha_{g|c} - 1} + \log \binom{\tilde{n}_{rs}^{(gc)} + \alpha_{g|c} - 1}{\alpha_{g|c} - 1} \\
&\quad = \log \frac{n_s^{(c)} + q_g\alpha_{g|c} - 1}{n_s^{(c)}} - \log \frac{n_{rs}^{(gc)} + \alpha_{g|c} - 1}{n_{rs}^{(gc)}} \\
&\quad \geq \log \frac{n_s^{(c)} + q_g\alpha_{g|c} - q_g}{n_s^{(c)}} - \log \frac{n_{rs}^{(gc)} + \alpha_{g|c} - 1}{n_{rs}^{(gc)}} \\
&\quad = \log \frac{n_s^{(c)}/q_g + \alpha_{g|c} - 1}{n_s^{(c)}/q_g} - \log \frac{n_{rs}^{(gc)} + \alpha_{g|c} - 1}{n_{rs}^{(gc)}} \\
&\quad \geq 0,
\end{aligned} \tag{C.7}$$

where in the last step we have made use of (C.6) and the fact that $\log((x + \alpha - 1)/x)$ is monotonically decreasing in x for all $x > 0$.

Now we observe that if there is any entry of $n^{(gc)}$ such that $n_{rs}^{(gc)} > 1$, then there must be an entry $n_{rs}^{(gc)} \geq n_s^{(c)}/q_g$, i.e., it is greater than or equal to the average for its column. We apply this observation repeatedly to decrement each non-zero entry of the table to 1 until $\tilde{n}_{rs}^{(gc)} = \min\left(n_{rs}^{(gc)}, 1\right)$, while at the same time ensuring that

$$\begin{aligned}
&\log \frac{\prod_s n_s^{(c)}!}{\prod_{rs} n_{rs}^{(gc)}!} + H(n^{(gc)}|n^{(c)}, q_g, \alpha_{g|c}) \\
&\qquad \geq \log \frac{\prod_s \tilde{n}_s^{(c)}!}{\prod_{rs} \tilde{n}_{rs}^{(gc)}!} + H(\tilde{n}^{(gc)}|\tilde{n}^{(c)}, q_g, \alpha_{g|c}).
\end{aligned} \tag{C.8}$$

This reduces the problem of showing the general inequality (C.4) to proving it for tables

$\tilde{n}$ whose entries are 0 or 1 only, which we can do as follows:

$$\begin{aligned}
&\log \frac{\prod_s \tilde{n}^{(c)}_s!}{\prod_{rs} \tilde{n}^{(gc)}_{rs}!} + H(\tilde{n}^{(gc)}|\tilde{n}^{(c)}, q_g, \alpha_{g|c}) \\
&\quad = \sum_s \Bigg[\log \tilde{n}^{(c)}_s! + \log \binom{\tilde{n}^{(c)}_s + q_g\alpha_{g|c} - 1}{q_g\alpha_{g|c} - 1} \\
&\qquad\qquad - \sum_r \log \binom{\tilde{n}^{(gc)}_{rs} + \alpha_{g|c} - 1}{\alpha_{g|c} - 1} \Bigg] \\
&\quad = \sum_s \big[\log(\tilde{n}^{(c)}_s + q_g\alpha_{g|c} - 1)! - \log(q_g\alpha_{g|c} - 1)! \\
&\qquad\qquad - \tilde{n}^{(c)}_s \log \alpha_{g|c} \big] \\
&\quad \ge \sum_s \left[\sum_{t=0}^{\tilde{n}^{(c)}_s - 1} \log(q_g\alpha_{g|c} + t) - \tilde{n}^{(c)}_s \log \alpha_{g|c} \right] && \text{(C.9a)} \\
&\quad \ge \sum_s \tilde{n}^{(c)}_s \big[\log(q_g\alpha_{g|c}) - \log \alpha_{g|c} \big] && \text{(C.9b)} \\
&\quad \ge \sum_s \tilde{n}^{(c)}_s \log q_g \ge q_g \log q_g, && \text{(C.9c)}
\end{aligned}$$

where in the final step we have made use of the fact that each of the q_g groups must contain at least one object, so there must be at least q_g nonzero entries in $n^{(gc)}$ and hence also in $\tilde{n}^{(gc)}$. We also observe that the inequalities (C.9a–C.9c) are saturated only when $\tilde{n}^{(c)}_s = 1$ for all s and $q_c = q_g$. These conditions together imply that the contingency table $n^{(gc)}$ must be diagonal, and hence that the labelings c and g are equivalent up to a permutation of their labels. The reverse conclusion, that $\text{RMI}_{\text{DM}}(c;g) = 1$ when c and g are equivalent up to a permutation, also follows since our measure is invariant under label permutations.

Finally, we note that if we instead normalize the reduced mutual information symmetrically according to

$$\text{NMI}^{(S)}_{\text{DM}}(c;g) = \frac{I_{\text{DM}}(c;g) + I_{\text{DM}}(g;c)}{I_{\text{DM}}(g;g) + I_{\text{DM}}(c;c)}, \tag{C.10}$$

then the results of this section also ensure that $\text{NMI}^{(S)}_{\text{DM}}(c;g) \le 1$ and that this bound is saturated only for c and g equivalent up to a permutation. This symmetric normalization may be more appropriate when comparing two labelings neither of which can be considered a ground truth.

C.14 Clustering and permutation invariance

In this paper we have focused on the comparison of different labelings of a set of objects, but the most common applications of the mutual information are actually to the comparison of *clusterings*, i.e., partitions of objects into some number q_g of (unlabeled) groups. One can easily represent a clustering by arbitrarily assigning integer labels $1 \ldots q_g$ to the groups and then recording the label of the group to which each object belongs, but the mapping from clusterings to labelings is not unique: here are $q_g!$ permutations of the labels that correspond to the same clustering. This means that the information cost of transmitting a labeling, as discussed in this paper, is larger than the information cost of transmitting a clustering. In the most extreme case, suppose that we want to transmit the unique clustering of n objects into n distinct groups, with a single object in each group. There are $n!$ possible labelings that represent this clustering, so the information cost to transmit any one of them is $H_0(g) = \log n!$ as in Eq. (3.6). Yet there is only a single clustering that places each object in its own group, so in principle the information cost should be $\log 1 = 0$. Thus the label-based approach grossly overestimates the true information cost in this case. As we argue in this appendix, however, the amount of the overestimate is a constant that plays no role in typical applications, and cancels completely from the mutual information itself, so in practice the measures described in this paper give correct and useful answers as is.

What is the actual information content of a clustering, not just of the labeling that represents it? To answer this question we adopt a notation that directly describes clusterings rather than labelings. For a given labeling g with q_g labels we define the equivalence class $\tilde{g}$ to be the set of all $q_g!$ variants of g obtained by permutations of the label values, including the original permutation g itself. By combining all these permutations into a single object, the equivalence class directly represents the clustering of which labeling g is a manifestation. With this definition we can adapt the encoding schemes for labelings described in this paper to give encoding schemes for clusterings.

Any encoding of labelings effectively defines a probability distribution over all labelings via $P(g) = e^{-H(g)}$. Since the schemes of this paper are all invariant under the $q_g!$ possible permutations of the labels, we can easily sum up the resulting probability weight over all labelings that represent a given clustering to find the induced probability distribution over clusterings:

$$P(\tilde{g}) = \sum_{q \in \tilde{g}} P(g) = q_g!\, P(g). \tag{C.11}$$

Under this distribution the cost to directly transmit the clustering $\tilde{g}$ independent of its label assignment is

$$H(\tilde{g}) = -\log P(\tilde{g}) = H(g) - \log q_g! \tag{C.12}$$

Thus, if we could find a way to transmit only the clustering we would realize an information savings of $\log q_g!$ compared with the transmission of an arbitrary labeling.

A practical way to achieve this is simply to agree upon a single unique labeling that will represent each possible clustering. Only these agreed labelings will be transmitted and no others. By definition this reduces the number of possible labelings by a factor of $q_g!$ and hence reduces the information by $\log q_g!$, as above.

To give an explicit example of such an encoding, we could stipulate that every labeling must have the following two properties:

1. Groups are labeled in order of increasing size, so that group 1 is the smallest and group q_g is the largest.

2. If two groups have the same size, the tie is broken by giving the smaller group label to the group that appears first in the ordered list of all objects.

For every clustering there is only one labeling that satisfies these rules, and any labeling that does not satisfy them can easily be converted into one that does. For example, $g = 33132112$ becomes 22321331.

If enforcement of the above rules is denoted by R, the information content of a clustering is

$$H(\tilde{g}) = H(g|R), \tag{C.13}$$

and with these definitions we can now explicitly calculate the information needed to transmit a clustering. As before, we transmit the clustering in three steps. In the first step we transmit the number of groups q_g. The fact that a labeling respects the rules R has no effect on q_g, so the information required for this step is unchanged from before: $H(q_g|R) = H(q_g)$.

In the second step we transmit the group sizes $n^{(g)}$, and here there is a change because rule 1 above implies that the group sizes must appear in non-decreasing order, and hence the possible values of $n^{(g)}$ are drawn only from the set of such non-decreasing candidates, a subset of the $\binom{n-1}{q_g-1}$ possible vectors that sum to n. We further note that not all of these non-decreasing vectors will occur with equal frequency. The number of ways one such vector can occur in our transmission process is equal to the number of unique starting vectors that can be permuted into the given non-decreasing form. If we define the multiplicity of the group sizes as

$$M_t = \big|\{r|n_r^{(g)} = t\}\big|, \qquad t = 1 \ldots q_g, \tag{C.14}$$

then there are $q_g!/\prod_t M_t!$ such permutations. So the probability that any individual one will occur is $(q_g!/\prod_t M_t!)/\binom{n-1}{q_g-1}$ and the information cost to transmit $n^{(g)}$ is minus the log of this probability:

$$H(n^{(g)}|q_g, R) = \log\left[\binom{n-1}{q_g-1}\frac{\prod_t M_t!}{q_g!}\right]. \tag{C.15}$$

In the third step of the transmission process we transmit the labeling itself, and here too the information cost is modified because of our rules. Whenever two groups of the same size are present, we know that the group appearing first must have the smaller group label because of rule 2 above and hence we need only consider labelings that satisfy this requirement. This leaves only a fraction $1/\prod_t M_t!$ of the original $n!/\prod_r n_r^{(g)}!$ labelings, giving an information cost of

$$H(g|n^{(g)}, R) = \log\frac{n!}{\prod_r n_r^{(g)}!\prod_t M_t!}. \tag{C.16}$$

Combining these terms, the total information cost to transmit the clustering is

$$\begin{aligned} H(\tilde{g}) &= H(g|R) \\ &= H(q_g|R) + H(n^{(g)}|q_g, R) + H(g|n^{(g)}, R) \\ &= H(q_g) + H(n^{(g)}) + \log\prod_t M_t! - \log q_g! \\ &\qquad + H(g|n^{(g)}) - \log\prod_t M_t! \\ &= H(g) - \log q_g! \end{aligned} \tag{C.17}$$

as expected.

Taking, for instance, our earlier example in which there are n groups of one object each, all groups have the same size, so by rule 2 above they are simply labeled in order of their appearance $123\ldots n$. This is the unique valid labeling with this set of group sizes, so setting $q_g = n$, the information cost is correctly given as $H(g) - \log q_g! = \log n! - \log n! = 0$.

The same discounted information cost also applies to the conditional entropy. Suppose we are given a candidate clustering denoted by equivalence class $\tilde{c}$ and represented as above by a unique labeling c within that class, such as the one obeying the rules R. Since our encoding schemes are invariant under label permutations, all labelings in $\tilde{c}$ are equally informative, including the one c that we have selected, and hence

$$H(\tilde{g}|\tilde{c}) = H(\tilde{g}|c). \tag{C.18}$$

Using the same argument as before, the conditional information cost of the clustering is then given by

$$H(\tilde{g}|c) = H(g|c) - \log q_g! \tag{C.19}$$

and hence the mutual information between two *clusterings* is given by

$$\begin{aligned} I(\tilde{c};\tilde{g}) &= H(\tilde{g}) - H(\tilde{g}|\tilde{c}) \\ &= \left[H(g) - \log q_g!\right] - \left[H(g|c) - \log q_g!\right] \\ &= H(g) - H(g|c) = I(c;g). \end{aligned} \tag{C.20}$$

Thus, the mutual information between clusterings is the same as between any corresponding pair of labelings. In practice, this means that we never need to consider mutual information measures between clusterings. Calculating the mutual information between labelings, as described in this paper, is more straightforward and gives the same result.

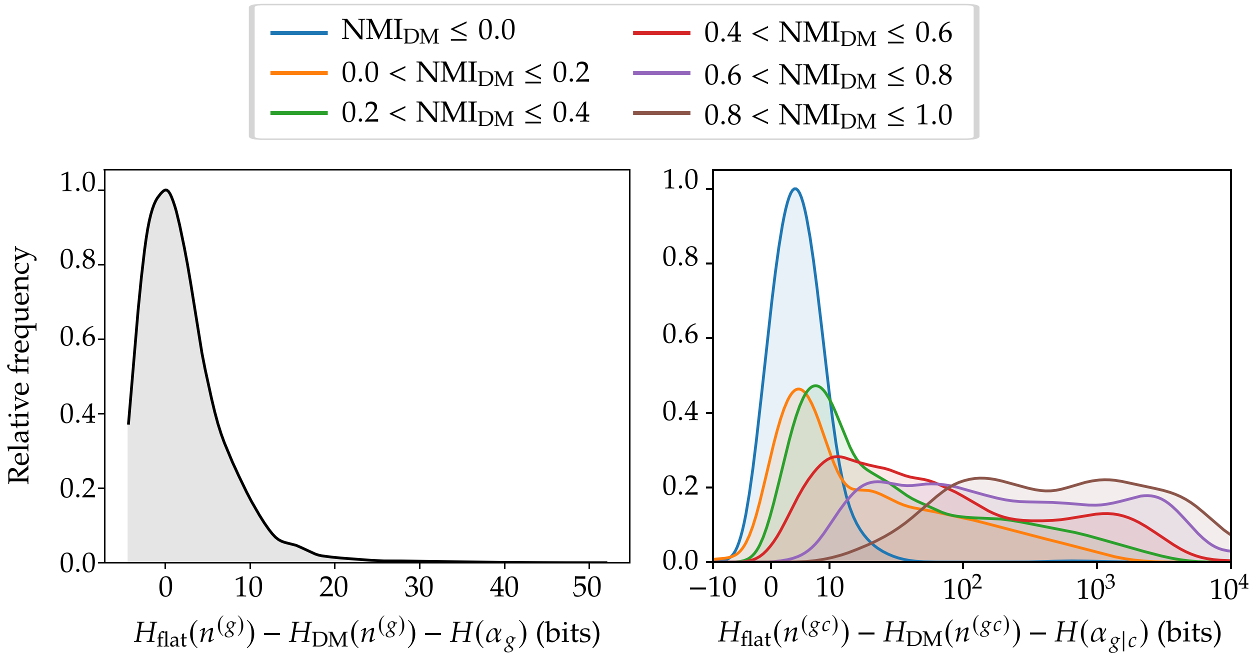
Using this clustering perspective we can also show that the encoding we propose in this paper is near optimal in the important case where $c = g$. All the encoding schemes we consider are invariant under label permutations, which implies that

$$H(g|g) = H(g|\tilde{g}) = H(\tilde{g}|\tilde{g}) + \log q_g! \geq \log q_g! \tag{C.21}$$

From Eqs. (3.31) and (3.33) our Dirichlet-multinomial encoding has cost

$$\begin{aligned} H_{\text{DM}}(g|g) &= H(q_g) + H(n^{(gg)}|n^{(g)}, \alpha_{g|g}) + H(g|c, n^{(gg)}) \\ &= \log n + q_g \log q_g. \end{aligned} \tag{C.22}$$

If we accept the cost $\log n$ of transmitting the number of groups q_g as a necessary price of doing business, this value for $H_{\text{DM}}(g|g)$ very nearly saturates the bound in Eq. (C.21), since the gap between $q_g \log q_g$ and $\log q_g!$ is only of order $\mathrm{O}(q_g)$. By contrast, the flat



(a)

(b)

Figure C.1: (a) The absolute change in the information cost of transmitting the vector of true group sizes $n^{(g)}$ between the standard flat encoding and the optimized Dirichlet-multinomial encoding. In contrast with Fig. 3.7, the information cost $H(\alpha_g) = 4$ bits for transmitting the Dirichlet-multinomial parameter is included in this comparison. (b) The absolute change in the information cost of transmitting the contingency table $n^{(gc)}$ between the standard flat encoding and the Dirichlet-multinomial encoding (including $H(\alpha_{g|c})$). The different curves show the distribution for different levels of similarity between the ground-truth and candidate labelings, as measured by the normalized mutual information. The horizontal scale is linear between −10 and 10 and logarithmic outside that range. In both panels the densities of cases are transformed and smoothed as in Fig. 3.7.

encoding is far from saturating the bound in this case, explaining its poorer performance in the important regime where $c \simeq g$. Equation (C.22) also helps explain a point made in Section 3.1.2.3, that it is rarely beneficial to constrain both the row and column sums of the contingency table, since the Dirichlet-multinomial encoding is already near-optimal while constraining only the columns.

C.15 Choosing and transmitting the Dirichlet-multinomial parameter

In computing the information costs $H(n^{(g)}|q_g, \alpha_g)$ and $H(n^{(gc)}|n^{(c)}, q_g, \alpha_{g|c})$ that appear in Eqs. (3.26) and (3.30) we have used the values of the Dirichlet-multinomial parameters α_g and $\alpha_{g|c}$ that minimize those costs. These values were found by numerical optimization, using golden-ratio search in the space of $\log \alpha$ with a starting bracket of $\alpha \in [10^{-3}, 10^3]$. Since the golden-ratio method converges exponentially, the complexity of this calculation grows logarithmically in the desired accuracy. In practice we converge to machine precision within about 50 iterations.

In Figs. 3.7 and 3.8 we compared the information costs of transmitting the group sizes $n^{(g)}$ and the contingency table $n^{(gc)}$ within the Dirichlet-multinomial encoding scheme and the standard (flat) encoding, but we neglected the cost of sending the value of α, which arguably means the comparison is not entirely fair. Assigning a cost to the transmission of α is somewhat delicate, since it is a continuous-valued parameter with a potentially infinite number of decimal digits, and hence its complete transmission would require an infinite amount of information. In practice, however, high accuracy is not needed to get most of the benefit of the Dirichlet-multinomial approach and we can use a small number of bits to transmit a value chosen from a finite set of possibilities without losing much. For example, by using four bits of information we can transmit a value chosen from the 16 possibilities $\alpha \in \{10^{-2}, 10^{-1.75}, 10^{-1.5}, \ldots, 10^{1.5}, 10^{1.75}\}$. In Figure C.1 we show the resulting difference in information cost between the Dirichlet-multinomial and flat encodings when this additional small cost is taken into account. As panel (a) shows, the cost of transmitting α_g does have a noticeable effect on the (already small) information to transmit $n^{(g)}$, the flat encoding now being favored in some cases, but this is usually not an issue, since the information cost of $n^{(g)}$ is not a large part of the total in most practical situations. As panel (b) shows, we retain the significant gains in the cost of transmitting the contingency table under the Dirichlet-multinomial scheme, even allowing for the cost of transmitting $\alpha_{g|c}$, especially in the common regime where $c \simeq g$.

Moreover, these concerns will not impact our final mutual information score at all if the same method is used to transmit both α_g and $\alpha_{g|c}$. Any costs that we include will cancel in the expression for the mutual information because

$$\begin{aligned}I_{\text{DM}}(c;g) &= I_0(c;g) + H(n^{(g)}|n, q_g, \alpha_g) + H(\alpha_g) \\ &\quad - \left[H(n^{(gc)}|n^{(c)}, q_g, \alpha_{g|c}) + H(\alpha_{g|c})\right] \\ &= I_0(c;g) + H(n^{(g)}|n, q_g, \alpha_g) - H(n^{(gc)}|n^{(c)}, q_g, \alpha_{g|c}). \end{aligned} \tag{C.23}$$

In practice, therefore, the cost of transmitting α plays no role in our calculation of the mutual information.

C.16 Benchmarking

In this appendix we give some additional details of our community detection tests.

C.16.1 Numbers of groups

As discussed in Section 3.2.4, the traditional (non-reduced) mutual information favors the $\gamma = 10$ generalized modularity maximization algorithm over other algorithms across many of the tests reported in Fig. 3.11. However, a closer inspection of the partitions found by this algorithm reveals that it drastically overestimates the number of communities. The number of groups inferred by each algorithm for each value of the mixing parameter μ is shown for $n = 3200$ in Fig. C.2. The traditional mutual information has a bias towards labelings with an excessive number of groups, which causes it to prefer the $\gamma = 10$ algorithm in this regime, while the reduced and adjusted mutual information measures prefer simple modularity maximization ($\gamma = 1$).

C.16.2 Community detection algorithms

We assess the performance of six common community detection algorithms on synthetic network examples. We use the algorithm implementations from the `igraph` library [35], except for the inference method, for which we use the `graph-tool` library [105]. The six algorithms are as follows.

1. **InfoMap:** InfoMap is an information theoretic approach that defines a compression algorithm for encoding a random walk on a network, based on which communities the walk passes through [118]. Different community labelings give rise to more or less efficient compression, as quantified by the so-called map equation, and the labeling with the highest efficiency is considered the best community division.

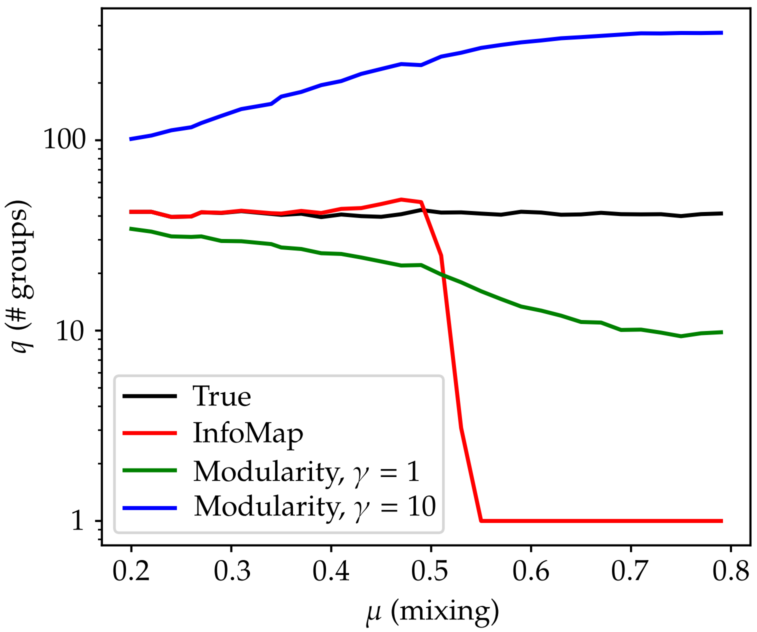


Figure C.2: The number of groups inferred by each of the six algorithms in Fig. 3.11 for LFR benchmark networks with $n = 3200$ nodes and a range of values of the mixing parameter μ. The true planted number of groups is shown in black. The InfoMap algorithm (red) generates an accurate number of groups for values of μ up to about 0.5, but beyond this point it erroneously places all nodes in a single group. Standard modularity maximization with resolution parameter $\gamma = 1$ (green) underestimates the number of groups, presumably because of the resolution limit on the detection of small groups, but not as severely as the number is overestimated when $\gamma = 10$ (blue).

2. **Modularity maximization:** Modularity is a quality function for network community divisions equal to the fraction of edges within communities minus the expected such fraction if edge positions are randomized while preserving node degrees. The labeling with the highest modularity is considered the best community division. Exact modularity maximization is NP-hard and usually intractable, but modularity can be maximized approximately using various heuristics, of which the most popular are agglomerative methods such as the Louvain and Leiden algorithms [18, 127], spectral methods [98], and simulated annealing [61, 89, 115]. In our tests we use simulated annealing where computationally feasible and the Leiden algorithm otherwise, these approaches giving the most consistent maximization of the modularity.

3. **Modularity with a resolution parameter:** Standard modularity maximization is known to suffer from a "resolution limit"—it cannot detect communities smaller than a certain threshold size [53], as discussed in Section 2.2.2. This can be remedied using a variant of modularity that includes a resolution parameter γ such that higher

values of γ cause the algorithm to prefer smaller communities [115]. Standard modularity maximization corresponds to $\gamma = 1$, but for comparison we also conduct tests with $\gamma = 10$ using the Leiden algorithm.

4. **Statistical inference:** Another popular approach to community detection makes use of model fitting and statistical inference. In this context the most commonly fitted model is the degree-corrected stochastic block model [69], which can be fitted using Bayesian methods to find the best community division [107].

5. **Walktrap:** Walktrap is an agglomerative algorithm in which initially separate nodes are iteratively combined into progressively larger communities in order from strongest to weakest connections, where strength is quantified in terms of the time for a random walk to reach one node from another [112].

6. **Labelprop:** The label propagation or "labelprop" algorithm initially places every node in its own community then iteratively updates the labels of randomly chosen nodes by majority vote among their network neighbors, breaking ties at random [113].

As discussed in Section 3.2.5, all of these algorithms perform reasonably well, but the best performers in our tests are InfoMap and the two variants of modularity maximization.

C.16.3 Results for the usual symmetric normalized mutual information

Figure C.3 shows the performance of the same six community detection methods as in Fig. 3.12, but measured using the standard, symmetrically normalized, non-reduced mutual information, which we denote by $I_0^{(S)}(c;g)$. By this measure, many of the methods appear to perform implausibly well, far beyond the detectability threshold visible in Fig. 3.12, in the regime where all methods should by rights fail. Note in particular the high scores achieved by the generalized modularity with $\gamma = 10$ by virtue of the excessive number of groups it generates.

C.16.4 LFR network generation

The LFR networks we use for benchmarking are generated using the procedure described in [77], which we summarize here.

1. **Draw a degree sequence** from a power-law distribution with exponent τ_1. Many networks have power-law degree distributions, typically with exponents between

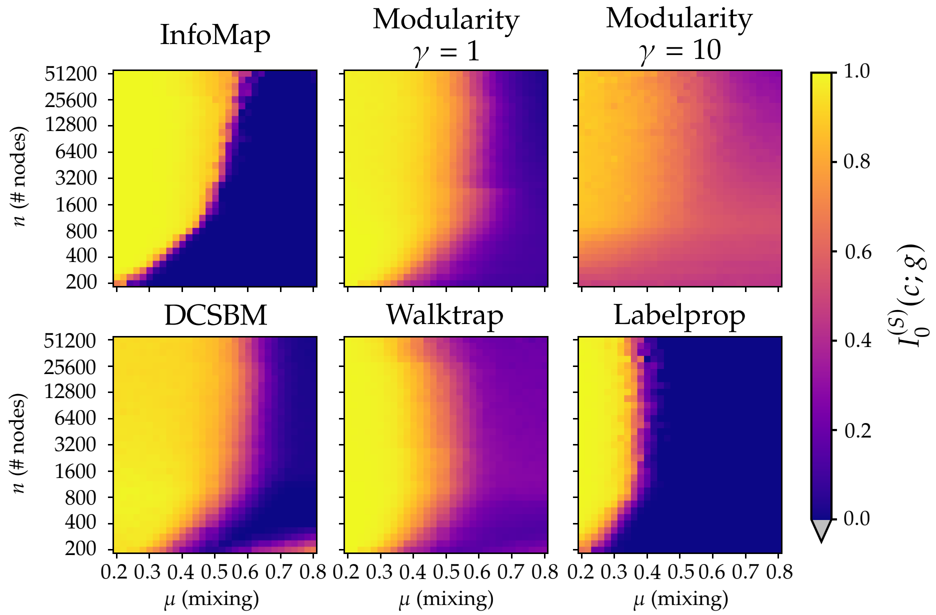


Figure C.3: Performance of each of the six community detection algorithms considered here, as quantified by the conventional symmetrically normalized non-reduced mutual information $I_0^{(S)}(c;g)$.

2 and 3 [21], and the LFR model exclusively uses power-law distributions. We use $\tau_1 = 2.5$, with average degree $\langle k \rangle = 20$ and maximum degree (which depends on graph size) $k_{\max} = n/10$. Empirically, however, our results do not seem to be very sensitive to these choices.

2. **Draw a set of community sizes** from a power-law distribution with exponent τ_2. Many networks are also found to have community sizes that approximately follow a power law, with typical exponents in the range from 1 to 2 [60, 32, 103, 77]. We use $\tau_2 = 1.5$ with a minimum community size of $s_{\min} = 20$ in all cases, while the maximum community size is set to $s_{\max} = \max(n/10, 100)$. Again, results were not particularly sensitive to these choices, provided they produce a valid distribution at all.

3. **Assign each node to a community** randomly, one node at a time, ensuring that the community chosen is large enough to support the added node's intra-community degree, given by $(1 - \mu)k$ where k is the total degree.

4. **Rewire the edges** attached to each node while preserving the node degrees, until the fraction of edges running between nodes in different communities is approximately μ.

The parameter values above are similar to those used for instance in [140]. As in that study, we find that algorithm performance is dictated primarily by the parameters n and μ, so it is these parameters that are varied our summary figures.

The LFR model is similar to a special case of the degree-corrected stochastic block model (DC-SBM) [69], and hence one might expect that inference-based community detection methods employing the latter model would perform well, perhaps even optimally, on LFR networks. Specifically, in the limit of an infinite number of sampled networks and perfect optimization of each community detection method, the final performance measure for any algorithm is given by the expectation value of the similarity $M(g, h[\boldsymbol{A}])$ between the ground truth LFR partition g and the partition $h[\boldsymbol{A}]$ of the LFR network $\boldsymbol{A}$ inferred using the algorithm, where the expectation is taken over the ensemble $P(\boldsymbol{A}, g|\theta)$ of LFR networks and partitions $\boldsymbol{A}, g$ generated using parameters θ (meaning μ, τ_1, τ_2, etc.). By using the LFR benchmark with parameters θ to compare the performance of community detection algorithms, we are therefore implicitly defining the "best" algorithm to be the one whose corresponding function $h[\boldsymbol{A}]$ optimizes $\sum_{\boldsymbol{A},g} P(\boldsymbol{A}, g|\theta)M(g, h[\boldsymbol{A}])$. If the similarity measure we choose is the "all or nothing" error function $M(g, c) = \delta(g, c)$, then the optimal community detection algorithm is trivially the one with

$$h[\boldsymbol{A}] = \operatorname*{argmax}_g P(\boldsymbol{A}, g|\theta) = \operatorname*{argmax}_g P(g|\boldsymbol{A}, \theta). \tag{C.24}$$

In other words, the optimal algorithm simply performs maximum a posteriori estimation under the model from which the network was generated. There is no explicit formula for the posterior probability under the LFR model, but to the extent that it is a special case of the DC-SBM, we might expect the DC-SBM (with appropriate priors) to give optimal results [109].

The LFR model, however, is not precisely a special case of the DC-SBM. In particular, the DC-SBM normally assumes a uniform distribution over community sizes, where the LFR model assumes a power law. Moreover, we are not using the crude all-or-nothing error function: our entire purpose in this paper is to develop mutual information measures that aggregate and weigh different modes of error in a sensible fashion. These differences, it appears, are enough to ensure that the DC-SBM does not perform the best in our testing.

Regardless, we emphasize that our use of the LFR benchmark in our analysis is simply for consistency with previous studies of network community detection methods [77, 75, 140]. The justification for our proposed similarity measure, on the other hand, is chiefly its theoretical merit over the conventional (symmetric, non-reduced) normalized mutual information, and is independent of the use of the LFR (or any other) benchmark.

C.17 Bias towards large numbers of groups in the traditional mutual information

To shed light on why the traditional mutual information Eq. 3.39 is biased towards an excessive number of groups, and how the reduced mutual information of Eq. 3.41 corrects this, consider the (extreme) example in which every object is placed in a group on its own, producing a candidate labeling $c = (1, \dots, n)$. Whatever the ground truth labeling g is, this candidate labeling clearly provides no information about it whatsoever, so we expect the mutual information to be zero. But this is not what we find. The contingency table in this case is

$$n_{rs}^{(gc)} = \begin{cases} 1 & \text{if } g_r = s, \\ 0 & \text{otherwise,} \end{cases} \tag{C.25}$$

so the conventional mutual information of Eq. (3.39) is

$$I_0(c;g) = \log \frac{n!}{\prod_s n_s^{(g)}!} = H_0(g). \tag{C.26}$$

This answer is not merely wrong; it is maximally so. The mutual information should take its minimum value of zero, but instead it takes the value $H_0(g)$, which is the maximum possible since $H_0(g)$ is an upper bound as we have said. The reason for this result is that in this case the contingency table itself uniquely defines g, so neglecting it puts the mutual information in error by an amount equal to the complete information cost of the ground truth. If we include the subleading terms on the other hand, this erroneous behavior disappears. Assuming for simplicity that the ground-truth groups are of equal size, the optimal concentration parameters are $\alpha_{g|c} = 0$ and $\alpha_g = \infty$, and the reduced mutual information of Eq. (3.41) becomes

$$\begin{aligned} I(c;g) &= I_0(c;g) - \log \left[\frac{n!}{\prod_r n_r!} (1/q_g)^n \right] - n \log q_g \\ &= 0. \end{aligned} \tag{C.27}$$

which is the correct answer.

Outside this special case, we observe from Eq. (3.41) that in the limit where the numbers of groups $q_g = q_c = q$ are held fixed and $n \to \infty$, the traditional mutual information grows as

$$I_0(c;g) \sim \mathrm{O}(n \log q) \tag{C.28}$$

while the correction term grows as

$$I(c;g) - I_0(c;g) \sim \mathrm{O}(q^2 \log n). \tag{C.29}$$

Thus, in this limit the subleading term is relevant when $\frac{n}{\log n} \lesssim \frac{q^2}{\log q}$, a condition that holds in many practical contexts, where the number of communities can grow as $\sqrt{n}$ or faster.

C.18 Adjusted mutual information

In this appendix we discuss the *adjusted mutual information* (AMI), an alternative version of the mutual information that addresses the typical measure's bias towards labelings with too many groups. We further discuss how this measure is related to the reduced mutual information we consider in Section 3.1 and how to interpret the AMI within the information theoretic framework.

For finite number of objects n, even a random labeling c will have positive mutual information with respect to any ground truth g in expectation: because the traditional mutual information is non-negative, fluctuations due to randomness will produce non-negative values only and hence their average will in general be positive [131, 145]. This seems counterintuitive; we would expect the average value for a random labeling to be zero.

We can solve this problem by subtracting off the expected value, thereby making the average zero by definition. To do this we must first specify how the expectation is defined—over what distribution of candidate labelings are we averaging? The conventional choice is to take the uniform distribution over labelings that share the same group sizes $n^{(c)}$ as the actual candidate c. This yields the *adjusted mutual information* of Vinh *et al.* [131]:

$$I_A(c;g) = I_0(c;g) - \left\langle I_0(c;g) \right\rangle_{\{c|n^{(c)}\}}, \tag{C.30}$$

where the expectation $\langle \ldots \rangle$ is over the relevant ensemble.

The adjusted mutual information can also be derived in a fully information-theoretic manner, as described in [101]. There it is shown that the subtracted term $\langle I_0(c;g) \rangle_{\{c|n^{(c)}\}}$ is precisely equal to the average cost of transmitting the contingency table when labelings are drawn from the uniform distribution. However, this distribution heavily favors contingency tables with relatively uniform entries, simply because there are many more labelings that correspond to uniform tables than to non-uniform ones. Real contingency tables, on the other hand, are often highly non-uniform, since applications of the mutual information focus on labelings that are somewhat similar to the ground truth (producing a non-uniform table). In such cases, the average used in the adjusted mutual information puts most of its weight on configurations that are very different from those that occur in reality, making it a poor representation of true information costs. The reduced mutual information considered in this thesis, by contrast, deliberately allows for non-uniform tables by drawing them from a Dirichlet-multinomial distribution and we argue that this is a strong reason to favor it over the adjusted mutual information. Nonetheless, in Section 3.2.3 we give results using both reduced and adjusted mutual information, and find fairly similar outcomes in the two cases.

Appendix D

Supplementary Material for Chapter 4

D.19 Data sets

The example data sets used in this chapter are summarized in Table 4.1 and divide into three broad categories: sports and games (six data sets), human social hierarchies (three data sets), and animal social hierarchies (six data sets). Here we provide some additional details on these data.

Sports and games: We consider both team competition (basketball, soccer) and individual competition (chess, Scrabble, tennis, video games). For the team sports we treat each team in each year as a different entity with its own assigned score s_i. Thus, for example, the England soccer team in 2015 is considered a different entity from the England soccer team in 2014. This reflects the fact that the composition of teams can change from season to season and with it the ranking of the team in comparison to others.

Two of the game data sets, for chess and Scrabble, were too large in their original forms to perform our full Bayesian analysis in a reasonable amount of time, so they were subsampled to reduce them to manageable size. We limited the chess data set to only those players who had participated in at least 200 games and then randomly selected 5% of those players. All others were removed from the data set. The Scrabble data set was similarly pared down by limiting it to players who had at least 100 games and then choosing a random 20% of those who remained.

Another issue with some of the game data is the presence of ties, which occur with moderate frequency in both chess and soccer. Although there do exist ranking models that allow for ties [114, 39], we avoid these in the present work for the sake of simplicity, and all our models assume that the only possible outcomes of a match are a win or a loss. To accommodate the chess and soccer data within this setting we remove all ties from the data, which amounts to 10–30% of matches in those data sets.

Human social hierarchies: A related issue arises in the "friends" data set, which details friend nominations among students in a US middle/high school. A substantial fraction of the nominations are reciprocal—two individuals each nominate the other as a friend [62, 12]. Such reciprocated nominations have been treated as ties in some previous analyses [100], but here again we simply remove them. Only unreciprocated friendships are recorded as a win for the person who receives the nomination.

For the faculty hiring data sets, the original source [31] included three data sets, for business schools, computer science departments, and history departments. We include only the first two of these in our analysis, purely to avoid cluttering the presentation, since the results for history departments very similar to the other two: we find $(\hat{\beta}, \hat{\alpha}) = (4.38, 0.01)$ for history, compared to $(4.36, 0.01)$ for business and $(4.25, 0.01)$ for computer science.

Animal hierarchies: Data on animal dominance hierarchies are copious: this has been an active field of research for at least sixty years. The data sets studied in this chapter come from a variety of sources, but particularly from DomArchive, a collection of 436 dominance interaction data sets compiled by Strauss *et al.* [125]. Data sets in the archive vary widely in size, but the sets we focus on are ones with a relatively large number of interactions per individual, which improves the statistics and helps reduce uncertainty on the fitted values of the model parameters.

D.20 Other measures of model performance

In the cross-validation results reported in Section 4.5 we quantify predictive performance of the various models by calculating the log-likelihood of the testing (held-out) data within the fitted model—see Fig. 4.4. This is not, however, the only way to measure performance. There are a number of other approaches in common use. In this appendix we describe some alternative performance metrics and investigate how our models size up when measured by these metrics. In general the results are similar to those presented in Section 4.5, but there are some differences in the details.

A simple way to quantify the predictive performance of a model is to count the number of times the model predicts the correct winner in the test data. As before, we start by fitting the model to the training portion of the data to obtain MAP estimates $\hat{\boldsymbol{s}}$ of the scores. Then, given those estimates, player i is considered favored to beat player j if $\hat{s}_i > \hat{s}_j$. The *accuracy* C of the model is defined to be the fraction of matches in the testing data where this prediction is born out:

$$C = \frac{\sum_{ij} W_{ij} \mathbf{1}_{\hat{s}_i > \hat{s}_j}}{\sum_{ij} W_{ij}}, \tag{D.1}$$

where W_{ij} is the number of times i beats j in the testing data, as previously, and $\mathbf{1}_x$ is the indicator function which is 1 if x is true and 0 otherwise.

Values of this accuracy measure are shown in Fig. D.1a for each of the models considered in this chapter for each of our data sets. As with our previous results for log-likelihood, we report performance relative to a baseline set by the standard Bradley-Terry model with a logistic prior, represented by the horizontal dashed line in the figure. Comparing with our earlier results from Fig. 4.4, the difference between models is smaller when measured in terms of accuracy than log-likelihood. For example, the minimum violations ranking performs quite poorly according to log-likelihood, but is comparable and sometimes better than our models in terms of accuracy. This may be because the minimum violations ranking is more directly tuned to solving this specific problem: by minimizing violations we precisely minimize the number of outcomes that are predicted incorrectly. On the other hand, the minimum violations algorithm does not reflect how confident we are in each outcome or any other aspect of the prediction task, and in this sense is inferior to other approaches.

Both the likelihood and accuracy measures are based on point estimates of model parameters $\hat{\boldsymbol{s}}$, $\hat{\alpha}$, and $\hat{\beta}$ but, as shown in Fig. 4.2, point estimates do not always do a good job of capturing the full posterior distribution $P(\boldsymbol{s}, \alpha, \beta | \boldsymbol{A}^{\text{train}})$, particularly in sparse data sets. To get around this issue we can calculate the average of the likelihood over the distribution of parameter values thus:

$$P(\boldsymbol{A}^{\text{test}} | \boldsymbol{A}^{\text{train}}) = \int P(\boldsymbol{A}^{\text{test}} | \boldsymbol{s}, \alpha, \beta) P(\boldsymbol{s}, \alpha, \beta | \boldsymbol{A}^{\text{train}}) \, d^n\boldsymbol{s} \, d\alpha \, d\beta. \tag{D.2}$$

In practice, this quantity can be estimated from a set of N samples of $(\boldsymbol{s}_k, \alpha_k, \beta_k)$ (with $k = 1 \ldots N$) drawn from the posterior $P(\boldsymbol{s}, \alpha, \beta | \boldsymbol{A}^{\text{train}})$, by computing the average

$$P(\boldsymbol{A}^{\text{test}} | \boldsymbol{A}^{\text{train}}) \simeq \frac{1}{N} \sum_{k=1}^{N} P(\boldsymbol{A}^{\text{test}} | \boldsymbol{s}_k, \alpha_k, \beta_k). \tag{D.3}$$

We can calculate this estimate from the same Monte Carlo samples we already generated, which we used previously to visualize the posterior distribution in Fig. 4.2. As our

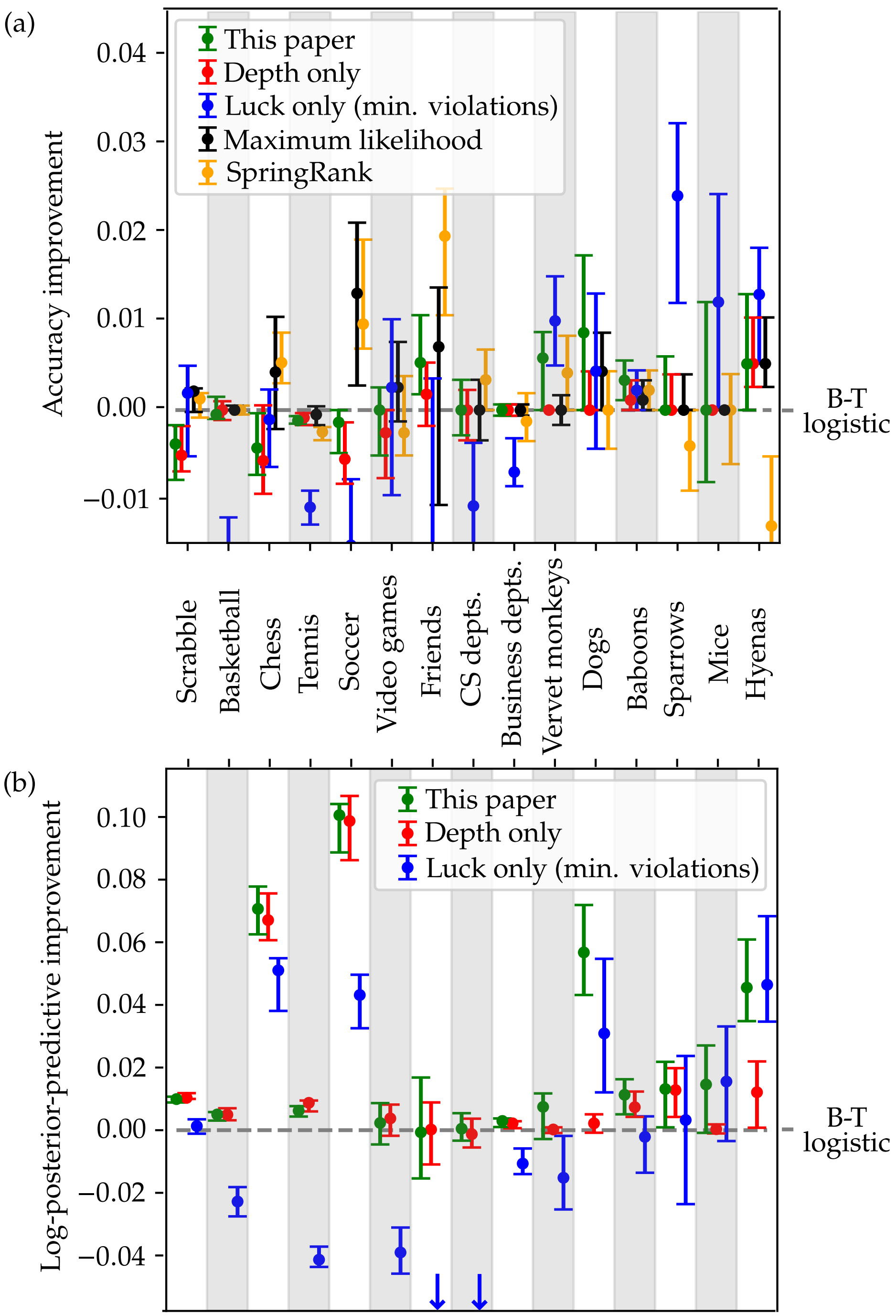


Figure D.1: Results from the same set of cross-validation tests shown in Fig. 4.4, but quantified using (a) accuracy and (b) log posterior-predictive probability, instead of log-likelihood. All results are measured relative to the Bradley-Terry model with a logistic prior, which is represented as the dashed horizontal line in each panel. Error bars represent upper and lower quartiles, estimated from at least 50 random repetitions of the cross-validation procedure in each case. The maximum likelihood and SpringRank models are not included in the lower comparison, since they are based on point estimates rather than Bayesian methods and hence one cannot calculate a posterior-predictive probability. Arrows indicate results that are off the scale.

measure of performance we then compute the *log posterior-predictive probability* per game

$$R = \frac{\log P(\boldsymbol{A}^{\text{test}}|\boldsymbol{A}^{\text{train}})}{\sum_{ij} W_{ij}}, \tag{D.4}$$

a fully Bayesian performance measure.

We plot this measure for a number of our models and data sets in Fig. D.1B. Note, however, that since the measure involves an integral over the posterior distribution of the scores, we cannot apply it to ranking methods that return point estimates of the scores only, rather than a full probability distribution, which in this case means the Bradley-Terry MLE and SpringRank, which are thus excluded from the figure. Among the remaining methods the full luck-plus-depth model of this chapter performs best, or equal-best, for every data set, by this measure.

D.21 Point estimates of parameters

To compute the log-likelihood and accuracy measures of predictive success we use point estimates of the model parameters and scores, which we compute one after the other: we estimate the expected posterior values of the parameters $\hat{\alpha}$, $\hat{\beta}$ from a simple average of the Monte Carlo samples, then we fix these values and compute the MAP values of the scores $\hat{\boldsymbol{s}}$ using a standard numerical optimization method. We could, alternatively, use the expected values of the scores, which would be easy to calculate from the samples, but we prefer MAP values since they give a more appropriate point of comparison with other approaches based on maximum probability estimates, such as the maximum likelihood fit to the Bradley-Terry model or the SpringRank algorithm.

One might imagine one could simplify the calculation by just jointly optimizing the posterior $P(\boldsymbol{s}, \alpha, \beta|\boldsymbol{A})$ over both the scores and parameters to define estimates

$$(\boldsymbol{s}^*, \alpha^*, \beta^*) \equiv \text{argmax}_{\boldsymbol{s},\alpha,\beta} P(\boldsymbol{s}, \alpha, \beta|\boldsymbol{A}). \tag{D.5}$$

We find, however, that this can give biased results by artificially inflating the value of the depth parameter β. This happens because the likelihood $P(\boldsymbol{A}|\boldsymbol{s}, \alpha, \beta)$ is a function of the product $\beta\boldsymbol{s}$ (see Eq. (4.13)), meaning that the value of the likelihood is unchanged if we increase β while simultaneously reducing all the scores by the same factor. Reducing $\boldsymbol{s}$ in this way increases the prior $P(\boldsymbol{s})$ (which is peaked at $\boldsymbol{s} = 0$) and so increases the posterior $P(\boldsymbol{s}, \alpha, \beta|\boldsymbol{A})$. Unchecked, this effect would send the joint maximum to $\beta^* \to \infty$, $\boldsymbol{s} \to 0$. The prior $P(\beta)$ somewhat mitigates this problem, but in practice the jointly fitted value β^*

is still unreasonably large: values for each of the data sets are shown in Table D.1.

D.22 Other measures of depth

In this chapter we measure depth of competition by the parameter β in our joint luck-plus-depth model, Eq. (4.13). This is not the only possible approach for quantifying depth, however, and in this appendix we discuss some alternative approaches and explain how they relate to similar ideas presented elsewhere.

As discussed in the section on depth of competition, our depth measure β counts the number of "levels of skill" between two typical players in a population, who in expectation have *a priori* score difference $s_i - s_j = 1$ (because of our choice of prior on s). An alternative, and common, way to define depth is as the number of levels between not the typical pair of players but the best and worst players, which is given by

$$\hat{\beta}_{\text{range}} = \hat{\beta}(\hat{s}_{\max} - \hat{s}_{\min}). \tag{D.6}$$

In the data sets studied here we find that the factor $\hat{s}_{\max} - \hat{s}_{\min}$ varies from about 2.5 to 4. The range tends to be larger when there are more competitors, presumably because outliers are more likely in large samples, and we regard this as downside of this measure, although in practice the depth order of our data sets does not change greatly between this measure and our own. Values of $\hat{\beta}_{\text{range}}$ are reported in Table D.1 for each of the data sets.

Our depth measure β is defined in the context of our full luck-plus-depth model, but in many cases, particularly for the sports data sets, there is no strong evidence of a nonzero luck parameter α. An alternative approach for quantifying depth in these cases is to use a depth-only model as in Eq. (4.9). Depth values calculated by fitting this model are given in Table D.1 and denoted β_0, which we refer to as "restricted depth." In practice these figures are not very different for those for β in cases (such as sports) where the value of α is small anyway, or more precisely when the posterior distribution in Figure 4.2 meaningfully intersects the $\alpha = 0$ axis, so that the zero-luck model is plausible. On the other hand, β and β_0 can differ substantially when the data support a significantly nonzero value of α. For example, the mice data set has an expected value of α around 0.25 with a posterior distribution that has considerable separation from $\alpha = 0$, and in this case we find a large difference between a value of $\hat{\beta} = 26.5$ and $\hat{\beta}_0 = 2.1$, the latter being more akin to the sports data than to the other animal hierarchies.

The restricted depth β_0 is closer in spirit to previous measures of depth that do not consider the element of luck, and the occurrence of large discrepancies with the value of β

in some data sets suggests that such previous measures might potentially be in error by a substantial margin. For applications where the element of luck is not an issue, however, the restricted depth could be useful as a simplification of our measure. It can be calculated relatively straightforwardly, to a good approximation, using the standard Bradley-Terry model with a logistic prior, a model we have recommended in the past. In our current analysis we have used Gaussian priors, but the logistic prior has some practical advantages in that it enables simple and fast iterative methods for computing MAP scores. In the most common version of this approach, one uses the unit logistic distribution $1/[(1+e^s)(1+e^{-s})]$ as prior with the standard ($\beta = 1$) Bradley-Terry model, which leads to an elegant iterative algorithm for calculating the scores [100]. The logistic prior, however, has variance $\frac{1}{3}\pi^2$, whereas our Gaussian prior has variance $\frac{1}{2}$, so, though the qualitative shape of the two distributions is similar, the logistic distribution has substantially greater width, by a factor of $\pi\sqrt{2/3}$. An alternative way to perform the same calculation is to shrink the width of the prior to be the same as the Gaussian, while simultaneously shrinking the width of the Bradley-Terry score function by the same factor, which is equivalent to choosing $\beta = \pi\sqrt{2/3} = 2.565$. This leaves the algorithm, and the resulting ranking, unchanged, and thus the iterative method with a logistic prior is equivalent to the depth-only model with $\beta = 2.565$.

Happily, this choice of β falls squarely in the middle of the range of values seen in Fig. 4.2 and in practice this approach has quite competitive performance, as shown in Fig. 4.4, where it is used as the baseline. On the other hand, there are plenty of cases where the value $\beta = 2.565$ is clearly misspecified, which is signaled by fitted scores whose variance does not match the width of the prior. This observation suggests that we could use the spread of the fitted scores as a heuristic measure of (restricted) depth and in practice this approach seems to work quite well. Quantifying the spread by its the standard deviation, we report figures for each of our data sets in Table D.1, and we find that there is good correlation between this standard deviation and the restricted depth $\hat{\beta}_0$ as calculated earlier. Given that the former is substantially easier to calculate than the latter, this could be a useful approach for calculations where accuracy and rigor are not at a premium.

We also note that MAP estimation in our depth-only model is equivalent to fitting the usual Bradley-Terry model with an L2 regularization, equivalent to a Gaussian prior. This correspondence suggests that one could infer a point estimate of the depth by tuning the strength of the L2 regularization by maximizing performance on some held-out validation data set. Like the joint MAP estimation of the depth and the scores discussed in Appendix D.21, however, this method displays a bias towards large depth values. Specifically, since validation performance is only assessed at the MAP point estimate of the scores, if that point scales as $\boldsymbol{s} \sim \beta^{-1}$ for large β the validation log-likelihood is largely unaffected, being proportional to βs. The Bayesian approach, by contrast, penalizes high values of β by effectively assessing performance integrated over the whole posterior distribution of scores, since large β values are sensitive to slight changes in these scores beyond the point estimate.

A quite different approach to measuring depth has been developed in the animal behavior literature, where the notion of "steepness" has gained currency in discussions of dominance hierarchies [41]. Steepness is most often defined through quantities known as "David's scores," which are measures of individual performance analogous to our fitted s_i [37]. The David's scores are defined as

$$\mathrm{DS}_i = w_i + \sum_j w_j P_{ij} - l_i - \sum_j l_j P_{ji} \tag{D.7}$$

where P_{ij} is the fraction of times that i beats j:

$$P_{ij} = \frac{A_{ij}}{A_{ij} + A_{ji}}, \tag{D.8}$$

and w_i and l_i are row and column sums of this matrix:

$$w_i = \sum_j P_{ij}, \qquad l_i = \sum_j P_{ji}. \tag{D.9}$$

De Vries *et al.* [41] propose normalizing the David's scores according to

$$\mathrm{NormDS}_i = \frac{\mathrm{DS}_i + \binom{n}{2}}{n}, \tag{D.10}$$

which vary between 0 and $n-1$, then the animals are ranked according to the resulting values. With the inferred rank order on the x-axis and the normalized David's score on the y-axis, the steepness of the hierarchy is then defined to be the slope S_{DS} of the ordinary line of best fit. A nice feature of this formulation is that the steepness runs from 0 to 1, with the value 1 being achieved in any hierarchy where all dominance interactions run from higher ranked to lower ranked individuals (zero violations).

Neumann and Fischer [96] have recently proposed a related measure that considers the slope S_{Elo} of the line of best fit between Elo scores for the competitors and their inferred ordinal ranking. Elo scores are essentially a sequential (time-dependent) version of a maximum likelihood fit to the Bradley-Terry model and so this definition is closer to the ideas considered in this chapter. Neumann and Fischer also incorporate Bayesian elements where certain aspects of the fitting process are randomized, such as the sequential order (if the true order is unknown) and the initial values of the ratings.

In Table D.1 we report values for a number of our data sets of S_{DS} (calculated using the R package `steepness` [80]) and S_{Elo} (calculated using the R package `EloSteepness` [95]). Overall, we find that the results are clearly correlated with the other measures shown in the table, although S_{DS} has trouble differentiating between the lower depth data sets. The Elo-based steepness S_{Elo} fares better and correlates quite well with the restricted depth $\hat{\beta}_0$, although the calculations are computationally demanding on account of the randomization and prove intractable for our larger data sets (as indicated by "–" in the table).

To complete our collection of measures of depth we also include in Table D.1 the parameter β_S that appears in the SpringRank model [40]. This parameter has not previously been used as a measure of depth but one can make an argument for its use in this way—see Appendix D.24.

Finally, we note in passing that there is an analogy between the depth parameter β and a notion of "temperature" for a data set. The form of the score function of Eq. (4.9) is precisely that of the Fermi-Dirac probability function of many-body physics, the probability of occupation at inverse temperature β of an energy level with energy s above the Fermi level. While we have not directly exploited this analogy here, it is a part of a broader correspondence between noise and unpredictability in statistics and temperature in physics in the same manner as the SBM-Ising model analogy described in Appendix A.5.

D.23 Depth as predictability

In Section 4.4 we observed that among our data sets the sports and games have lower depth compared to the social hierarchies, and we speculated that this was because a high-depth sport would not be as interesting to watch: at high depth a typical pair of competitors will be very unevenly matched and there will be little suspense about who is going to win. In other words, high depth should result in high predictability of outcomes. In this appendix we test this hypothesis by calculating various measures of predictability.

A natural measure of predictability is the same log-likelihood that we studied in our section on predicting wins and losses. The log-likelihood of a data set is equal to minus the description length of the outcomes of the matches in that set, given the fitted model. That is, it is equal to the amount of information it would take to communicate the outcomes

		Measures of depth								Luck	
	Data set	$\hat{\beta}$	β^*	$\hat{\beta}_{\text{range}}$	$\hat{\beta}_0$	std($\hat{\boldsymbol{s}}_L$)	S_{DS}	S_{Elo}	$\hat{\beta}_S$	$\hat{\alpha}$	α^*
Sports/games	Scrabble	0.68	3.13	2.43	0.60	0.64	0.00	–	2.24	0.09	0.00
	Basketball	1.01	10.79	3.66	0.83	0.61	0.01	0.48	2.32	0.13	0.02
	Chess	1.17	4.73	4.21	1.04	0.91	0.00	–	2.85	0.07	0.12
	Tennis	1.44	1.98	5.88	1.34	0.72	0.00	–	2.67	0.04	0.00
	Soccer	1.73	6.23	4.97	1.58	1.02	0.00	–	4.00	0.04	0.00
	Video games	1.77	17.53	5.12	1.55	1.10	0.02	0.62	2.95	0.07	0.05
Human	Friends	3.54	10.36	9.88	2.80	1.16	0.00	–	5.23	0.05	0.00
	CS departments	4.25	15.42	12.11	3.88	1.88	0.01	0.78	4.46	0.01	0.00
	Business depts.	4.36	13.72	11.73	4.07	2.25	0.14	0.84	4.07	0.01	0.01
Animal	Vervet monkeys	6.01	30.39	17.07	3.57	2.23	0.40	0.85	4.34	0.07	0.07
	Dogs	8.74	33.29	24.82	3.76	2.03	0.25	0.93	3.65	0.11	0.09
	Baboons	13.19	18.61	39.04	9.37	4.38	0.05	0.95	5.63	0.02	0.02
	Sparrows	22.92	63.89	69.68	8.68	3.62	0.50	0.91	7.72	0.02	0.01
	Mice	26.48	59.48	72.29	2.10	1.35	0.31	0.72	3.22	0.25	0.24
	Hyenas	100.58	168.48	246.42	9.83	4.00	0.30	0.95	8.15	0.02	0.02

Table D.1: Inferred parameter values for the data sets considered in Section 4.4. From left to right: $\hat{\beta}$ is expected depth, β^* is the jointly optimized MAP depth as in Eq. (D.5), $\hat{\beta}_{\text{range}}$ is depth between the best and worst player as in Eq. (D.6), $\hat{\beta}_0$ is restricted depth as inferred in the depth-only ($\alpha = 0$) model, std($\hat{\boldsymbol{s}}_L$) is the standard deviation of the MAP scores within the logistic-prior model, S_{DS} is the steepness measure of de Vries *et al.* [41], S_{Elo} is the steepness measure of Neumann and Fischer [96], $\hat{\beta}_S$ is the maximum likelihood estimate of the parameter β_S in the SpringRank model [40], $\hat{\alpha}$ is the expected luck, and α^* is the jointly optimized MAP estimate of the luck.

to a receiver who already knows the fitted model. Higher information (more negative log-likelihood) implies more unpredictable outcomes. Completely random outcomes (matches decided by the toss of a coin) would give a log-likelihood of −1 per match (in log-base-2 units), while completely predictable ones would give zero.

Previously, we plotted the log-likelihood relative to the baseline set by the standard Bradley-Terry model, but in the present context we are interested in the absolute value. Figure D.2A shows the absolute value for each of our data sets, arranged in order of increasing depth β. As the figure shows, the low-depth sports on the left are indeed quite unpredictable and none of our models perform much better than chance at predicting outcomes (log-likelihood per match is close to −1). Some of the methods we compare against, notably the maximum likelihood Bradley-Terry and minimum violation ranking, fall well short even of random guesses, as indicated by the arrows at the bottom of the figure. As depth increases, however, outcomes generally become more predictable, and the deepest animal hierarchies have a log-likelihood approaching zero, meaning outcomes are nearly perfectly predictable.

There are some exceptions to this trend, most notably the mice data set which, as seen in Fig. 4.2, has a large element of luck ($\hat{\alpha} \simeq 0.25$). This introduces substantial randomness into the matches, despite the high depth, and greatly decreases predictability.

We can shed further light on predictability by calculating the average amount of information needed to describe matches that are truly drawn from our model. That is, we consider two players whose scores s_i are drawn from our normal prior with variance $\frac{1}{2}$, so that the difference of their scores is normally distributed with variance 1, and we assume that the probability of i beating j is given exactly by $p_{ij} = f_{\alpha\beta}(s_i - s_j)$, Eq. (4.13), for some values of α and β that we specify. Then the average information needed to describe the outcome of the match is given by the standard entropy function for a Bernoulli random variable

$$H[p_{ij}] = -p_{ij} \log p_{ij} - (1 - p_{ij}) \log(1 - p_{ij}). \tag{D.11}$$

Then, writing $s = s_i - s_j$ and integrating, the average entropy per match over matches between many random pairs of players is

$$S_{\alpha\beta} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} H\left[f_{\alpha\beta}(s)\right] e^{-s^2/2}\, ds. \tag{D.12}$$

Unfortunately, this integral does not seem to have a closed-form solution, but it can be evaluated numerically. Figure D.3 shows a modified version of Fig. 4.2 from the main

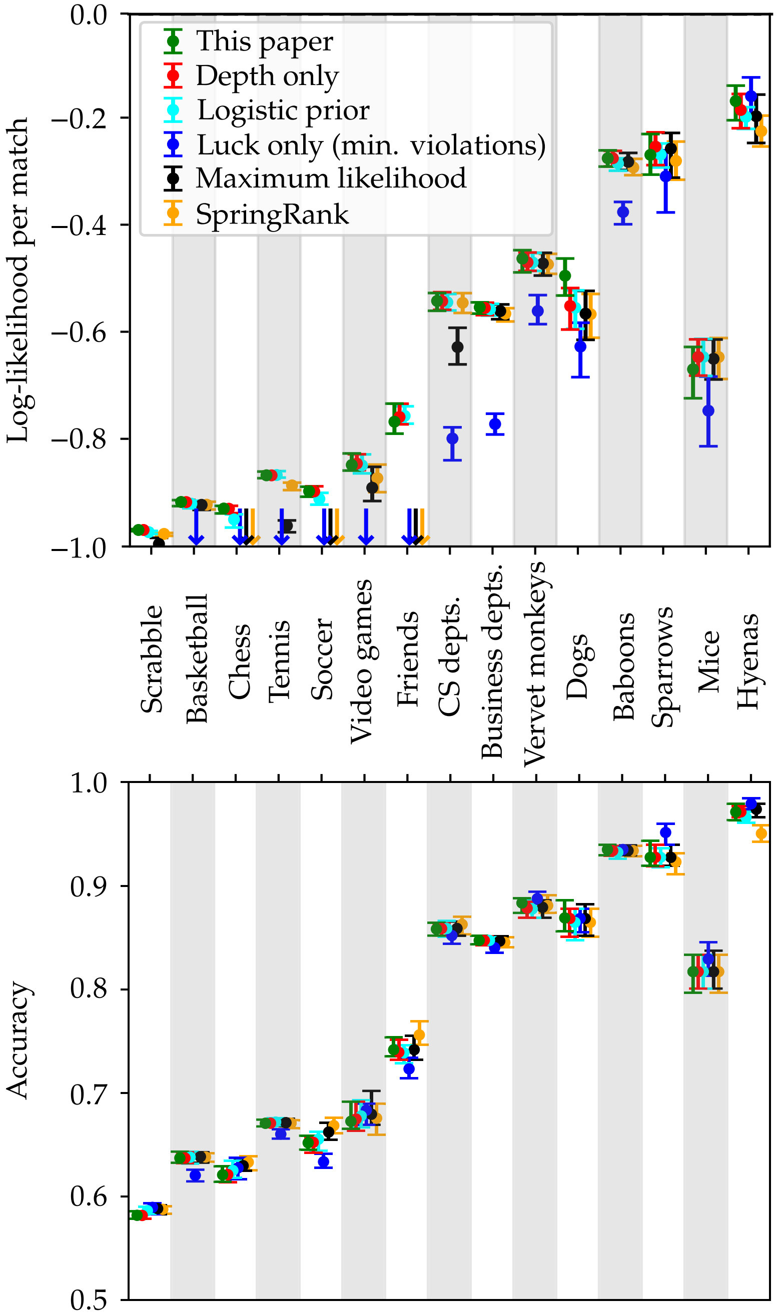


Figure D.2: Absolute log-likelihood and accuracy values per match in the cross-validation tests of Fig. 4.4. This figure differs from Fig. 4.4 in showing absolute values rather than values relative to the Bradley-Terry model with logistic prior.

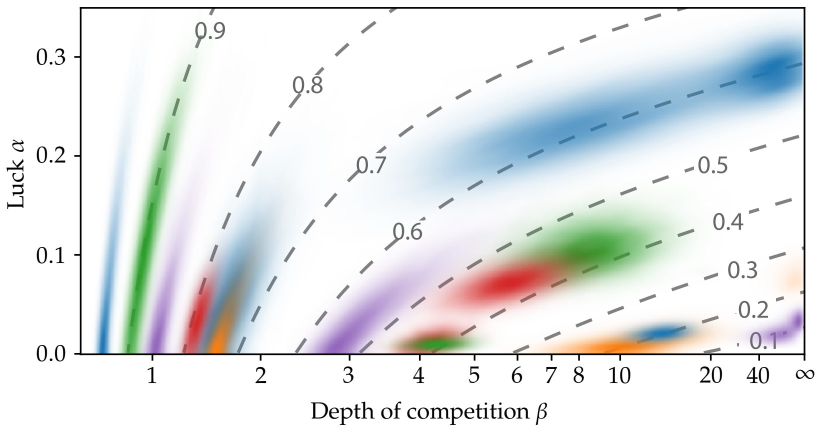


Figure D.3: The data sets of Fig. 4.2 with dashed lines representing the contours of average entropy per match. Low entropy indicates confidence about the outcome of a match; high entropy indicates unpredictability.

chapter, representing the posterior probability distribution of α, β for our various data sets, with superimposed lines representing the contours of the average entropy. As the figure shows, the entropy is higher for lower depth and for higher luck, as we would expect, since both increase the unpredictability of outcomes. We also note that the posterior distributions of individual data sets appear to follow the contour lines quite closely, arcing upward and to the right. This occurs because the entropy is by definition equal to minus the log-likelihood, and our prior on α and β is slowly varying by construction, so the posterior is also slowly varying along the contour lines of constant likelihood. The contour lines are calculated as averages over outcomes drawn from the fitted model, whereas the probability clouds in the figure represent real-world data, so the two are not precisely comparable. But to the extent that the data are well described by the model we would expect them to agree and hence for the clouds to follow the contours in the plot. This also means that, while some of the clouds in the figure are quite extended, indicating substantial uncertainty about the values of α and β, they are narrow in the direction perpendicular to the contours, meaning that we have high confidence about the value of the log-likelihood. This is reflected in Fig. D.2a, where we see that the uncertainty on our estimates of the log-likelihood is quite modest.

D.24 SpringRank

Among the various approaches to ranking considered in this chapter, SpringRank [40] is a recent and novel approach based on a physical analogy to the behavior of a network of masses and springs. In this appendix we make some observations on the method and how it relates to the Bradley-Terry model, which forms the foundation for the other methods we consider.

In SpringRank the likelihood of observing a directed network $\boldsymbol{A}$ is given by a product of Poisson distributions over all possible directed edges:

$$P(\boldsymbol{A}|\boldsymbol{s},\beta_S,c)=\prod_{ij}\frac{r_{ij}^{A_{ij}}}{A_{ij}!}e^{-r_{ij}}, \tag{D.13}$$

with the expect number of directed edges $i \to j$ given by

$$r_{ij}=ce^{-\frac{1}{2}\beta_S(s_i-s_j-1)^2}, \tag{D.14}$$

for given scores $\boldsymbol{s}$, inverse temperature β_S, and a "sparsity" parameter c. Equation (D.13) can be rewritten as

$$\begin{aligned}P(\boldsymbol{A}|\boldsymbol{s},\beta_S,c)&=\prod_{i<j}\frac{r_{ij}^{A_{ij}}}{A_{ij}!}e^{-r_{ij}}\,\frac{r_{ji}^{A_{ji}}}{A_{ji}!}e^{-r_{ji}}\\&=\prod_{i<j}\frac{(r_{ij}+r_{ji})^{A_{ij}+A_{ji}}e^{-(r_{ij}+r_{ji})}}{(A_{ij}+A_{ji})!}\frac{(A_{ij}+A_{ji})!}{A_{ij}!A_{ji}!}\left(\frac{r_{ij}}{r_{ij}+r_{ji}}\right)^{A_{ij}}\left(\frac{r_{ji}}{r_{ij}+r_{ji}}\right)^{A_{ji}}\\&=\prod_{i<j}\frac{m_{ij}^{\bar{A}_{ij}}e^{-m_{ij}}}{\bar{A}_{ij}!}\binom{\bar{A}_{ij}}{A_{ij}}\frac{1}{[1+e^{-2\beta_S(s_i-s_j)}]^{A_{ij}}[1+e^{-2\beta_S(s_j-s_i)}]^{A_{ji}}},\end{aligned} \tag{D.15}$$

where $m_{ij}=r_{ij}+r_{ji}$ and $\bar{A}_{ij}=A_{ij}+A_{ji}$ is an element of the adjacency matrix $\bar{\boldsymbol{A}}$ of the undirected network of matches.

Equation (D.15) is equal to the likelihood of generating an undirected network $\bar{\boldsymbol{A}}$ of matches and then separately choosing the directions of the edges, i.e., the winners of the matches:

$$P(\boldsymbol{A}|\boldsymbol{s},\beta_S,c)=P(\bar{\boldsymbol{A}}|\boldsymbol{s},\beta_S,c)\,P(\boldsymbol{A}|\boldsymbol{s},\beta_S,\bar{\boldsymbol{A}}), \tag{D.16}$$

where the probability of the undirected network is another product of Poisson distributions:

$$P(\bar{\boldsymbol{A}}|\boldsymbol{s},\beta_S,c)=\prod_{i<j}\frac{m_{ij}^{\bar{A}_{ij}}e^{-m_{ij}}}{\bar{A}_{ij}!} \tag{D.17}$$

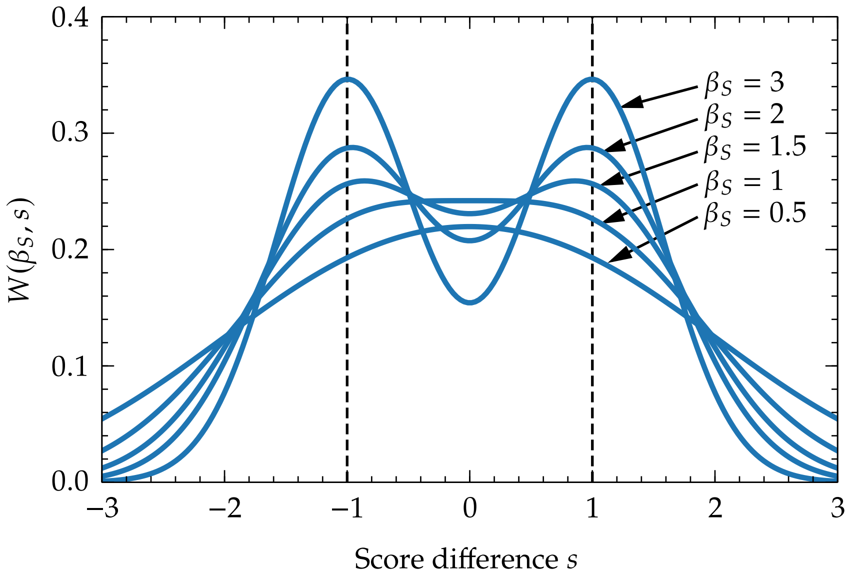


Figure D.4: The function $W(\beta_S, s)$ of Eq. (D.21) plotted against s, for various values of β_S as indicated.

and

$$P(\boldsymbol{A}|\boldsymbol{s}, \beta_S, \bar{\boldsymbol{A}}) = \prod_{i<j} \binom{\bar{A}_{ij}}{A_{ij}} \frac{1}{[1 + e^{-2\beta_S(s_i - s_j)}]^{A_{ij}} [1 + e^{-2\beta_S(s_j - s_i)}]^{A_{ji}}}. \tag{D.18}$$

(It is straightforward to confirm that the latter is correctly normalized for $A_{ij} = 0 \ldots \bar{A}_{ij}$ and $A_{ji} = \bar{A}_{ij} - A_{ij}$.)

But Eq. (D.18) is identical to the likelihood for the model studied in this chapter, Eqs. (4.3) and (4.13),

$$P(\boldsymbol{A}|\boldsymbol{s}, \alpha, \beta, \bar{\boldsymbol{A}}) = \prod_{i<j} \binom{\bar{A}_{ij}}{A_{ij}} f_{\alpha\beta}(s_{ij})^{A_{ij}} f_{\alpha\beta}(s_{ji})^{A_{ji}}, \tag{D.19}$$

if we choose $\alpha = 0$ and $\beta = 2\beta_S$. (The binomial coefficient accounts for the number of ways of assigning directions A_{ij} to the $\bar{A}_{ij}$ undirected edges.) This observation suggests that we might use β_S as a a measure of the (restricted) depth of a hierarchy, and indeed we observe a correlation between the maximum likelihood value $\hat{\beta}_S$ and our own restricted depth parameter β_0, as shown in Table D.1.

However, it is the other term, Eq. (D.17), that particularly distinguishes SpringRank from the other models we have considered. This term, which measures the likelihood that the set of observed matches occurs at all, has no equivalent in the Bradley-Terry model and related models. The quantity m_{ij}, which is the expected number of matches between i and j, can be rewritten in the form

$$m_{ij} = M \frac{W(\beta_S, s_i - s_j)}{\sum_{i<j} W(\beta_S, s_i - s_j)}, \tag{D.20}$$

where

$$W(\beta_S, s) = \sqrt{\frac{\beta_S}{8\pi}} \bigl[e^{-\frac{1}{2}\beta_S(s-1)^2} + e^{-\frac{1}{2}\beta_S(s+1)^2} \bigr]. \tag{D.21}$$

(Note that $W(\beta_S, s)$ is symmetric in s so the sign of the score difference in Eq. (D.20) has no effect.) In this formulation the parameter M controls the total number of (undirected) edges in the network and the (properly normalized) probability density $W(\beta_S, s_i - s_j)$ controls how they are distributed given the scores s_i. Figure D.4 shows the form of $W(\beta_S, s)$ for various choices of β_S. For $\beta_S \leq 1$ there is a single peak at $s = 0$ so that interactions are preferentially between evenly matched players, but above $\beta_S = 1$ the function becomes bimodal and increasingly peaked around $s = \pm 1$, so that players with a score difference near 1 are more likely to interact.

It is arguably a disadvantage of the SpringRank model that the same parameter β_S controls both the depth of competition via Eq. (D.18) and the distribution of matches via Eq. (D.20). Conceptually these are separate processes, and one could make an argument for a model in which they were controlled by separate parameters, although we have not taken that approach here—we use the model as originally defined for the sake of consistency.

In our cross-validation tests we use the maximum likelihood point estimate for the value of β_S, in keeping with the other models we study. We note, however, that De Bacco *et al.* [40], in their original work on SpringRank, used different values of β_S depending on whether the results were scored using log-likelihood or accuracy, choosing in each case the value that gave the best performance according to the measure used. However, unlike the definition given in [40], our definition of the accuracy is independent of the choice of β_S.

Finally, we note that the original specification of the SpringRank model also included an optional Gaussian prior on the scores. We have not adopted this prior in our tests, since we find that it tends to diminish the performance of the method.

Bibliography

1. Unpublished network compiled and labeled by V. Krebs.

2. College football records. URL. Accessed: 2025-05-17.

3. Online chess match data from lichess.com. URL. Accessed: 2023-10-07.

4. Scrabble tournament records. https://*www*.cross-tables.com/. Accessed: 2023-10-07.

5. Super Smash Bros. Melee head to head records. URL. Accessed: 2023-10-07.

6. Emmanuel Abbe. Community detection and stochastic block models: recent developments. *Journal of Machine Learning Research*, 18(177):1–86, 2018.

7. Lada A. Adamic and Natalie Glance. The political blogosphere and the 2004 US election. In *Proceedings of the WWW-2005 Workshop on the Weblogging Ecosystem*, New York, 2005. Association of Computing Machinery.

8. A. Agresti. *Categorical Data Analysis*. Wiley, New York, 1990.

9. Alessia Amelio and Clara Pizzuti. Correction for closeness: Adjusting normalized mutual information measure for clustering comparison. *Computational Intelligence*, 33:579–601, 2017.

10. L.N.F. Ana and A.K. Jain. Robust data clustering. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pages II–II, 2003.

11. Samin Aref, Hriday Chheda, and Mahdi Mostajabdaveh. The Bayan algorithm: Detecting communities in networks through exact and approximate optimization of modularity. Preprint arxiv:2209.04562, 2022.

12. Brian Ball and M. E. J. Newman. Friendship networks and social status. *Network Science*, 1:16–30, 2013.

13. Michael J. Barber. Modularity and community detection in bipartite networks. *Phys. Rev. E*, 76:066102, Dec 2007.

14. Alexander Barvinok and Jay Hartigan. Maximum entropy Gaussian approximations for the number of integer points and volumes of polytopes. *Advances in Applied Mathematics*, 45:252–289, 2010.

15. Michael Betancourt. A conceptual introduction to Hamiltonian Monte Carlo. Preprint arxiv:1701.02434, 2017.

16. Elisângela L. S. Bezerra, Isabel C. Machado, and Marco A. R. Mello. Pollination networks of oil-flowers: a tiny world within the smallest of all worlds. *Journal of Animal Ecology*, 78(5):1096–1101, 2009.

17. Neli Blagus, Lovro Šubelj, and Marko Bajec. Self-similar scaling of density in complex real-world networks. *Physica A: Statistical Mechanics and its Applications*, 391(8):2794–2802, 2012.

18. Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *J. Stat. Mech.*, 2008:P10008, 2008.

19. S. Boccaletti, M. Ivanchenko, V. Latora, A. Pluchino, and A. Rapisarda. Detection of complex networks modularity by dynamical clustering. *Phys. Rev. E*, 75:045102, 2007.

20. R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39:324–345, 1952.

21. Guido Caldarelli. *Scale-Free Networks*. Oxford University Press, Oxford, 2007.

22. George T. Cantwell and M. E. J. Newman. Message passing on networks with loops. *Proc. Natl. Acad. Sci. USA*, 116:23398–23403, 2019.

23. Alessio Cardillo, Jesús Gómez-Gardeñes, Massimiliano Zanin, Miguel Romance, David Papo, Francisco del Pozo, and Stefano Boccaletti. Emergence of network features from multiplexity. *Scientific Reports*, 3:1344, 2013.

24. François Caron and Arnaud Doucet. Efficient Bayesian inference for generalized Bradley-Terry models. *Journal of Computational and Graphical Statistics*, 21:174–196, 2012.

25. Manuela Cattelan. Models for paired comparison data: A review with emphasis on dependent data. *Statistical Science*, 27:412–433, 2012.

26. Marie-Liesse Cauwet, Olivier Teytaud, Hua-Min Liang, Shi-Jim Yen, Hung-Hsuan Lin, I-Chen Wu, Tristan Cazenave, and Abdallah Saffidine. Depth, balancing, and limits of the Elo model. *Proceedings of the 2015 IEEE Conference on Computational Intelligence and Games 2015*, 2015.

27. Claudia Cavallaro, Vincenzo Cutello, and Mario Pavone. Effective heuristics for finding small minimal feedback arc set even for large graphs. In *itaDATA*, 2023.

28. Shuo Chen and Thorsten Joachims. Modeling intransitivity in matchup and comparison data. In *Proceedings of the ninth acm international conference on web search and data mining*, pages 227–236, 2016.

29. Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

30. Fan Chung and L. Lu. Connected components in random graphs with given degree sequences. *Annals of Combinatorics*, 6:125–145, 2002.

31. Aaron Clauset, Samuel Arbesman, and Daniel B. Larremore. Systematic inequality and hierarchy in faculty hiring networks. *Science Advances*, 1:e1400005, 2015.

32. Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70:066111, 2004.

33. Clyde H. Coombs, Robyn M. Dawes, and Amos Tversky. *Mathematical psychology: An elementary introduction.* Prentice-Hall, 1970.

34. Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley, New York, 2 edition, 2006.

35. Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.

36. Leon Danon, Jordi Duch, Albert Diaz-Guilera, and Alex Arenas. Comparing community structure identification. *J. Stat. Mech.*, 2005:P09008, 2005.

37. H. A. David. *The Method of Paired Comparisons*. Griffin, London, 2 edition, 1988.

38. R. R. Davidson and D. L. Solomon. A Bayesian approach to paired comparison experimentation. *Biometrika*, 60:477–487, 1973.

39. Roger R. Davidson. On extending the Bradley-Terry model to accommodate ties in paired comparison experiments. *Journal of the American Statistical Association*, 65:317–328, 1970.

40. Caterina De Bacco, Daniel B. Larremore, and Cristopher Moore. A physical model for efficient ranking in networks. *Science Advances*, 4:eaar8260, 2018.

41. H. De Vries, Jeroen M. G. Stevens, and Hilde Vervaecke. Measuring and testing the steepness of dominance hierarchies. *Animal Behaviour*, 55:585–592, 2006.

42. Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Inference and phase transitions in the detection of modules in sparse networks. *Phys. Rev. Lett.*, 107:065701, 2011.

43. Persi Diaconis and Bradley Efron. Testing for independence in a two-way table: New interpretations of the chi-square statistic. *Annals of Statistics*, 13:845–874, 1985.

44. Byron E. Dom. An information-theoretic external cluster-validity measure. In Adnan Darwiche and Nir Friedman, editors, *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, pages 137–145, San Francisco, CA, 2002. Morgan Kaufmann.

45. Manlio De Domenico, Vincenzo Nicosia, Alexandre Arenas, and Vito Latora. Structural reducibility of multilayer networks. *Nature communications*, 6(1):6864, 2015.

46. C. A. A. Duineveld, Paul Arents, and Bonnie M. King. Log-linear modelling of paired comparison data from consumer tests. *Food Quality and Preference*, 11(1-2):63–70, 2000.

47. Martin Dyer, Ravi Kannan, and John Mount. Sampling contingency tables. *Random Structures & Algorithms*, 10:487–506, 1997.

48. Martin E. Dyer and Alan M. Frieze. The solution of some random np-hard problems in polynomial expected time. *Journal of Algorithms*, 10(4):451–489, 1989.

49. Felix Effenberger. A primer on information theory with applications to neuroscience. In *Computational Medicine in Data Mining and Modeling*, pages 135–192. Springer, 2013.

50. Paul Erdős and Alfréd Rényi. On random graphs. *Publicationes Mathematicae*, 6:290–297, 1959.

51. Edwin Fong and Chris C. Holmes. On the marginal likelihood and cross-validation. *Biometrika*, 107(2):489–496, 2020.

52. L. R. Ford, Jr. Solution of a ranking problem from binary comparisons. *American Mathematical Monthly*, 64(8):28–33, 1957.

53. Santo Fortunato and Marc Barthélemy. Resolution limit in community detection. *Proc. Natl. Acad. Sci. USA*, 104:36–41, 2007.

54. Santo Fortunato and Darko Hric. Community detection in networks: A user guide. *Physics Reports*, 659:1–44, 2016.

55. Mathias Franz, Emily McLean, Jenny Tung, Jeanne Altmann, and Susan C Alberts. Self-organizing dominance hierarchies in a wild primate population. *Proceedings of the Royal Society B*, 282:20151512, 2015.

56. Corrado Gini. *Variabilità e mutabilità: contributo allo studio delle distribuzioni e delle relazioni statistiche.[Fasc. I.]*. Tipogr. di P. Cuppini, 1912.

57. Michelle Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA*, 99:7821–7826, 2002.

58. Benjamin H. Good, Yves-Alexandre de Montjoye, and Aaron Clauset. Performance of modularity maximization in practical contexts. *Phys. Rev. E*, 81:046106, 2010.

59. I. J. Good. On the application of symmetric Dirichlet distributions and their mixtures to contingency tables. *Annals of Statistics*, 4:1159–1189, 1976.

60. R. Guimerà, L. Danon, A. Díaz-Guilera, F. Giralt, and A. Arenas. Self-similar community structure in a network of human interactions. *Phys. Rev. E*, 68:065103, 2003.

61. Roger Guimerà, Marta Sales-Pardo, and Luis A. N. Amaral. Modularity from fluctuations in random graphs and complex networks. *Phys. Rev. E*, 70:025101, 2004.

62. M. T. Hallinan and W. N. Kubitschek. The effect of individual and structural characteristics on intransitivity in social networks. *Social Psychology Quarterly*, 51:81–92, 1988.

63. David R. Hunter. MM algorithms for generalized Bradley-Terry models. *Annals of Statistics*, 32:384–406, 2004.

64. Maximilian Jerdee, Alec Kirkley, and M. E. J. Newman. Improved estimates for the number of non-negative integer matrices with given row and column sums. *Proc. R. Soc. London A*, 480:20230470, 2024.

65. Maximilian Jerdee, Alec Kirkley, and M. E. J. Newman. Mutual information and the encoding of contingency tables. *Phys. Rev. E*, 110:064306, Dec 2024.

66. Maximilian Jerdee, Alec Kirkley, and MEJ Newman. Normalized mutual information is a biased measure for classification and community detection. *arXiv:2307.01282*, 2023.

67. Maximilian Jerdee and M. E. J. Newman. Luck, skill, and depth of competition in games and social hierarchies. *Science Advances*, 10(45):eadn2654, 2024.

68. Mart Jürisoo. International men's football results from 1872 to 2023. URL. Accessed: 2023-10-07.

69. Brian Karrer and M. E. J. Newman. Stochastic blockmodels and community structure in networks. *Phys. Rev. E*, 83:016107, 2011.

70. Alec Kirkley. Spatial regionalization based on optimal information compression. *Communications Physics*, 5:249, 2022.

71. Alec Kirkley. Inference of dynamic hypergraph representations in temporal interaction data. Preprint arxiv:2308.16546, 2023.

72. Alec Kirkley and M. E. J. Newman. Representative community divisions of networks. *Communications Physics*, 5:40, 2022.

73. Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, and Mason A. Porter. Multilayer networks. *Journal of Complex Networks*, 2:203–271, 2014.

74. Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguñá. Hyperbolic geometry of complex networks. *Phys. Rev. E*, 82:036106, 2010.

75. A. Lancichinetti and S. Fortunato. Community detection algorithms: A comparative analysis. *Phys. Rev. E*, 80:056117, 2009.

76. Andrea Lancichinetti and Santo Fortunato. Consensus clustering in complex networks. *Scientific reports*, 2(1):336, 2012.

77. Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Phys. Rev. E*, 78:046110, 2008.

78. Amy N. Langville and Carl D. Meyer. *Who's #1? The Science of Rating and Ranking*. Princeton University Press, Princeton, 2013.

79. Nathan Lauga. NBA games data. URL. Accessed: 2023-10-07.

80. David Leiva and Han de Vries. *Testing steepness of dominance hierarchies*, 2022. R package, version 0.3-0.

81. Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, 2005. Association of Computing Machinery.

82. M. O. Lorenz. Methods of measuring the concentration of wealth. *Publications of the American Statistical Association*, 9:209–219, 1905.

83. R. D. Luce. *Individual Choice Behavior: A Theoretical Analysis*. John Wiley, New York, 1959.

84. David Lusseau, Karsten Schneider, Oliver J. Boisseau, Patti Haase, Elisabeth Slooten, and Steve M. Dawson. The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. Can geographic isolation explain this unique trait? *Behavioral Ecology and Sociobiology*, 54:396–405, 2003.

85. Rahul Makhijani and Johan Ugander. Parametric models for intransitivity in pairwise rankings. In *The World Wide Web Conference*, pages 3056–3062, 2019.

86. Laruent Massoulié. Community detection thresholds and the weak Ramanujan property. In *Proceedings of the 46th Annual ACM Symposium on the Theory of Computing*, pages 694–703, New York, 2014. Association of Computing Machinery.

87. Ralph H. Masure and Warder C. Allee. The social order in flocks of the common chicken and the pigeon. *The Auk*, pages 306–327, 1934.

88. Aaron F. McDaid, Derek Greene, and Neil Hurley. Normalized mutual information to evaluate overlapping community finding algorithms. Preprint arxiv:1110.2515, 2011.

89. A. Medus, G. Acuña, and C. O. Dorso. Detection of community structures in networks via global optimization. *Physica A*, 358:593–604, 2005.

90. Stanley Milgram. The small world problem. *Psychology Today*, 2:60–67, 1967.

91. James Moody. Race, school integration, and friendship segregation in America. *Am. J. Sociol.*, 107:679–716, 2001.

92. Cristopher Moore. The computer science and physics of community detection: Landscapes, phase transitions, and hardness. Preprint arxiv:1702.00467, 2017.

93. Elchanan Mossel, Joe Neeman, and Allan Sly. Reconstruction and estimation in the planted partition model. *Probability Theory and Related Fields*, 162:431–461, 2015.

94. Radford M. Neal. MCMC using Hamiltonian dynamics. In Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng, editors, *Handbook of Markov Chain Monte Carlo*, pages 113–162. Chapman and Hall, New York, 2011.

95. Christof Neumann. *EloSteepness: Bayesian dominance hierarchy steepness via Elo rating and David's scores*, 2023. R package, version 0.5.0.

96. Christof Neumann and Julia Fischer. Extending Bayesian Elo-rating to quantify the steepness of dominance hierarchies. *Methods in Ecology and Evolution*, 14:669–682, 2023.

97. M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E*, 74:036104, 2006.

98. M. E. J. Newman. Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA*, 103:8577–8582, 2006.

99. M. E. J. Newman. Ranking with multiple types of pairwise comparisons. *Proc. R. Soc. London A*, 478:20220517, 2022.

100. M. E. J. Newman. Efficient computation of rankings from pairwise comparisons. *Journal of Machine Learning Research*, 24:238, 2023.

101. M. E. J. Newman, George T. Cantwell, and Jean-Gabriel Young. Improved mutual information measure for clustering, classification, and community detection. *Phys. Rev. E*, 101:042304, 2020.

102. Günce Keziban Orman, Vincent Labatut, and Hocine Cherifi. Qualitative comparison of community detection algorithms. *Communications in Computer and Information Science*, 167:265–279, 2011.

103. Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, 2005.

104. Leto Peel, Daniel B. Larremore, and Aaron Clauset. The ground truth about metadata and community detection in networks. *Science Advances*, 3:e1602548, 2017.

105. Tiago P. Peixoto. The graph-tool python library. *figshare*, 2014.

106. Tiago P. Peixoto. Hierarchical block structures and high-resolution model selection in large networks. *Phys. Rev. X*, 4:011047, 2014.

107. Tiago P. Peixoto. Nonparametric Bayesian inference of the microcanonical stochastic block model. *Phys. Rev. E*, 95:012317, 2017.

108. Tiago P. Peixoto. Merge-split markov chain monte carlo for community detection. *Phys. Rev. E*, 102:012305, Jul 2020.

109. Tiago P Peixoto. Revealing consensus and dissensus between network partitions. *Physical Review X*, 11:021003, 2021.

110. Tiago P. Peixoto and Alec Kirkley. Implicit models, latent compression, intrinsic biases, and cheap lunches in community detection. *Physical review. E*, 108(2-1):024309–024309, 2023.

111. R. L. Plackett. The analysis of permutations. *Journal of the Royal Statistical Association C*, 24:193–202, 1975.

112. Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. In Pinar Yolum, Tunga Güngör, Fikret S. Gürgen, and Can C. Özturan, editors, *Proceedings of the 20th International Symposium on Computer and Information Sciences*, volume 3733 of *Lecture Notes in Computer Science*, pages 284–293, New York, 2005. Springer.

113. Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E*, 76:036106, 2007.

114. P. V. Rao and L. L. Kupper. Ties in paired-comparison experiments: A generalization of the Bradley-Terry model. *Journal of the American Statistical Association*, 62:194–204, 1967.

115. Joerg Reichardt and Stefan Bornholdt. Statistical mechanics of community detection. *Phys. Rev. E*, 74:016110, 2006.

116. Maria A. Riolo, George T. Cantwell, Gesine Reinert, and M. E. J. Newman. Efficient method for estimating the number of communities in a network. *Phys. Rev. E*, 96:032310, 2017.

117. William Robertie. *Inside Backgammon*, 2(1):3–4, 1980.

118. Martin Rosvall and Carl T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci. USA*, 105:1118–1123, 2008.

119. Jeff Sackmann. ATP tennis data. URL. Accessed: 2023-10-07.

120. Tatang Mitra Setia and Carel P. Van Schaik. The response of adult orangutans to flanged male long calls: inferences about their function. *Folia Primatologica*, 78(4):215–226, 2007.

121. Matthew J Silk, Michael A Cant, Simona Cafazzo, Eugenia Natoli, and Robbie A McDonald. Elevated aggression is associated with uncertainty in a network of dog dominance interactions. *Proceedings of the Royal Society B*, 286:20190536, 2019.

122. R. Solomonoff and A. Rapoport. Connectivity of random nets. *Bulletin of Mathematical Biophysics*, 13:107–117, 1951.

123. Ulrich Stelzl, Uwe Worm, Maciej Lalowski, Christian Haenig, Felix H. Brembeck, Heike Goehler, et al. A human protein-protein interaction network: a resource for annotating the proteome. *Cell*, 122(6):957–968, 2005.

124. E. D. Strauss and K. E. Holekamp. Social alliances improve rank and fitness in convention-based societies. *Proceedings of the National Academy of Sciences*, 116:8919–8924, 2019.

125. Eli D Strauss, Alex R DeCasien, Gabriela Galindo, Elizabeth A Hobson, Daizaburo Shizuka, and James P Curley. DomArchive: A century of published dominance data. *Philosophical Transactions of the Royal Society B*, 337:20200436, 2022.

126. Henri Theil. *Henri Theil's contributions to economics and econometrics: econometric theory and methodology. Vol. I*, volume 1. Springer Science & Business Media, 1992.

127. Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9:5233, 2019.

128. J. Richard Udry, Peter S. Bearman, and Kathleen Mullan Harris. National Longitudinal Study of Adolescent Health, 1997. This research uses data from Add Health, a program project directed by Kathleen Mullan Harris and designed by J. Richard Udry, Peter S. Bearman, and Kathleen Mullan Harris at the University of North Carolina at Chapel Hill, and funded by grant P01–HD31921 from the Eunice Kennedy Shriver National Institute of Child Health and Human Development, with cooperative funding from 23 other federal agencies and foundations. Special acknowledgment is due Ronald R. Rindfuss and Barbara Entwisle for assistance in the original design. Information on how to obtain the Add Health data files is available on the Add Health website (https://www.cpc.unc.edu/addhealth). No direct support was received from grant P01–HD31921 for this analysis.

129. Robert E. Ulanowicz and Donald L. DeAngelis. Network analysis of trophic dynamics in south florida ecosystems. *US Geological Survey Program on the South Florida Ecosystem*, 114(45):234, 2005.

130. Chloé Vilette, Tyler Bonnell, Peter Henzi, and Louise Barrett. Comparing dominance hierarchy methods using a data-splitting approach with real-world data. *Behavioral Ecology*, 31:1379–1390, 2020.

131. Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11:2837–2854, 2010.

132. Yuchung J. Wang and George Y. Wong. Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82:8–19, 1987.

133. D. J. Watt. Relationship of plumage variability, size and sex to social dominance in Harris' sparrows. *Animal Behaviour*, 34:16–27, 1986.

134. Duncan J. Watts. Networks, dynamics, and the small world phenomenon. *Am. J. Sociol.*, 105:493–592, 1999.

135. Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

136. John T. Whelan. Prior distributions for the Bradley-Terry model of paired comparisons. Preprint arxiv:1712.05311, 2017.

137. J. G. White, E. Southgate, J. N. Thompson, and S. Brenner. The structure of the nervous system of the nematode Caenorhabditis elegans. *Phil. Trans. R. Soc. London B*, 314:1–340, 1986.

138. Cait M Williamson, Becca Franks, and James P Curley. Mouse social network dynamics and community structure are associated with plasticity-related brain gene expression. *Frontiers in Behavioral Neuroscience*, 10:152, 2016.

139. Xiaoran Yan, Cosma Rohilla Shalizi, Jacob E. Jensen, Florent Krzakala, Cristopher Moore, Lenka Zdeborová, Pan Zhang, and Yaojia Zhu. Model selection for degree-corrected block models. *J. Stat. Mech.*, 2014:P05007, 2014.

140. Zhao Yang, René Algesheimer, and Claudio J. Tessone. A comparative analysis of community detection algorithms on artificial networks. *Scientific Reports*, 6:30750, 2016.

141. Tzu-Chi Yen and Daniel B. Larremore. Community detection in bipartite networks with stochastic block models. *Phys. Rev. E*, 102:032309, Sep 2020.

142. W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.

143. Ernst Zermelo. Die Berechnung der Turnier-Ergebnisse als ein Maximumproblem der Wahrscheinlichkeitsrechnung. *Mathematische Zeitschrift*, 29:436–460, 1929.

144. Lizhi Zhang and Tiago P. Peixoto. Statistical inference of assortative community structures. *Physical Review Research*, 2(4):043271, 2020.

145. Pan Zhang. Evaluating accuracy of community detection using the relative normalized mutual information. *J. Stat. Mech.*, 2015:P11006, 2015.

146. Yunpeng Zhao, Elizaveta Levina, and Ji Zhu. Consistency of community detection in networks under degree-corrected stochastic block models. *Annals of Statistics*, 40:2266–2292, 2011.