Analysis of Algorithms in Learning Theory

and

Network Analysis of Knowledge Bases

BY

DIMITRIOS IOANNIS DIOCHNOS
Ptychion (National and Kapodistrian University of Athens, Hellas) 2004
M.Sc. (National and Kapodistrian University of Athens, Hellas) 2007

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Mathematics
in the Graduate College of the
University of Illinois at Chicago, 2013

Chicago, Illinois

Defense Committee:

György Turán, Chair and Advisor
Dhruv Mubayi
Lev Reyzin
Jan Verschelde
Robert H. Sloan, Computer Science, University of Illinois at Chicago

To my parents,

Ioannis and Alexandra,

and my sister Aggeliki

Acknowledgments

So early in the final manuscript, and yet, this part of the thesis is the last one written. One tries to recall the years, the people, the effort, and the time spent towards the pursuit of a PhD degree. It is certainly an interesting process to recall some details. Through that process I can immediately realize that the six years that I spent at the University of Illinois at Chicago (UIC), and more broadly in the United States, were very rewarding. For that I am really grateful to many people that I had the privilege to interact with, primarily in Chicago and mainly at UIC. Gratitude is something that can not be expressed fully in words but the least I can do is to try to say a *thank you* here. I had a really generous support of research assistanships throughout my years at UIC. The work that appears in this thesis was supported by NSF Grants CCF-0916708 and IIS-0747369.

Let me start with my adviser, Professor György Turán. I met Gyuri in August of 2007 when I was just starting my studies at UIC and Gyuri was teaching a course in complexity theory. From that class, as well as from the classes that followed later on, it became apparent to me that Gyuri is precisely the model of a Professor that I wanted to have while I was a student. Tremendous clarity and right on the spot explanations for many interesting concepts. The courses were really eye-opening and I still find myself going back to the notes of these courses from time to time. But beyond that, I am really grateful to him for showing me trust in my endeavors and giving me the chance to be his student and work on subjects that I found interesting[1]. Gyuri was always there for me when I needed his advice. Even if we were miles apart, I knew that I could send him an email and an answer would be imminent[2].

I also had the opportunity to follow Gyuri in Hungary during his Sabbatical in the Fall semester of 2011. Performing research without any teaching obligations, I was also given the chance to get to know Budapest, Szeged, researchers, Hungarians and their hospitality, the everyday life, the food, and the culture of Hungary. Not surprisingly, Gyuri wanted to make me feel even more like home and I was also introduced to Greeks who live in Hungary. To my surprise though, this interaction was an incentive for me to dive into the modern history of Greece and discover details about the bonds between Greece and Hungary in modern times. All in all, these four months in Hungary were four months with very vivid memories.

Going one step beyond, Gyuri, Rozsa, and Julia treated me as if I was a member of their family with their hospitality both in Chicago as well as in Budapest. There were many occasions that they invited me to their home and I had a wonderful time each and every time. Not only that, but I had the privilege to eat superb food and desserts made by Rozsa and Julia. In fact, Rozsa made me realize that there is actually a recipe in which, not only I can eat chickpeas, but it can become one of my very favorite dishes! Thank you very much Rozsa for the recipe; this will follow me for a lifetime not only as an excellent dish, but as an example of “never saying never” as well - even after about 30 years of consecutive trials and failures!

[1] An example of the freedom that I had is my work that appeared in [37], which was also supported by the NSF Grant CCF-0916708 but is not part of this thesis.

[2] At this point I can not help but recall times at 2 or 3 o’clock in the morning when I would send an email to Gyuri related to research and this could trigger a discussion with a chain of emails that could last for an hour or so, and nevertheless, both of us would still have to go to UIC early next day.

Let me now come to Professor Robert Sloan. Professor Sloan has acted as my second adviser while I was pursuing my PhD. I may not have attended any course taught by Professor Sloan, but I was lucky enough to have the opportunity to work and collaborate with him in about half of the subjects studied in my thesis. Professor Sloan was also always there for me when I needed him or his advice. We had many interesting discussions for problems that arise in learning theory as well as for problems that arise in reasoning, knowledge bases, and more broadly in artificial intelligence. This small paragraph does not do justice to the respect and the gratitude that I have for his personality and the help, both in terms of research as well as in terms of administrative matters at UIC, that he has provided me constantly throughout these years. I hope though, that I have fulfilled to a big extent the expectations that he had in the projects that we have worked together. Apart from Gyuri, I also want to thank Professor Sloan for the trust that he showed towards myself and I ended up having a generous support from their mutual grant these years.

I was also very fortunate that Professors Dhruv Mubayi, Lev Reyzin, and Jan Verschelde made me the honor to participate in my thesis committee. I am grateful to their comments, corrections, and suggestions not only for the content of this thesis, but to a bigger extent to their human side and advice that they have given me during my studies at UIC.

Pursuing a PhD degree required a lot of effort. In fact, to a very big extent effort that was unrelated to research. Perhaps this is a little bit more apparent to international students who have to go through some additional processes and procedures as individuals in a foreign country. As I am writing these lines I am in Edinburgh, making all the preparations for the beginning of my postdoc position. These days really remind me of my first days in the US back in 2007 when I was trying to take care of my obligations and be ready for the beginning of the journey which is called PhD. Coming back to the point of this paragraph, I am now certain that without the help and support of my family and friends I would not have been able to accomplish the journey. For that I owe them a big thank you.

In particular, Jim, Katie, Sophia F, Paul R, Jenna, Aivry, Holly, Ed, Michael, Ella, Emily, Andy, Marc, Wanyu, and Karen, all of them embraced me in their country and made me feel like home. I am grateful to them for contributing to a balanced social life for me throughout these years. Moreover, through them, I had the opportunity to live days, nights, and events as Americans do, and I am not sure if I would have had such an opportunity otherwise. However, I am pretty sure that we will have the chance to meet numerous times again in the future in different corners of this planet. I will be very happy when these days come to life!

I also met many Greeks in Chicago. Without Vangeli my beginning in the US would have been more difficult and I am grateful that I met him and his family. It was lovely that we organized, from time to time, “Greek” meetings with Vangeli, Despina, Maria K, and Sam, all of whom have Greek or Cypriot roots and were affiliated in one way or another with the Department of Mathematics, Statistics, and Computer Science at UIC. Katerina, Maria G, Vilelmini, Anna, Lena, George G, Varvara, and George L, were among the first ones I met, had a wonderful time exploring Chicago with them, and I am really happy that we still keep in touch and meet whenever we have the opportunity. A special thank you goes to Katerina with whom I was staying at the same building and had so many times conversations and food together while she was still in Chicago. Despina and Aleks have a special place in my heart. I knew that I could always rely on them, get an extra hand of help when I needed it, and of course I am grateful for all the time that we spent together. Same goes to Thomai, Jeremey and their little daughter Dafni. I am very happy that I met them, grateful for all their help, and I am really looking forward to meeting them next time that all of us will be in Greece. Through Thomai I met many people at the Hellenic-American Academy in Deerfield and it was great fun, as well as my honor, to participate with her, Taki, Mario, Niko G, Helen, Sophia K, Marianthi, Vicky, Kosta P, and Niko R in a theatrical play. Of course it would have been a great omission from my side not to mention Alexandro, Giota P, Vasili, Dina, and their families who also embraced me in Chicago. They made my life much easier and I was given the opportunity to follow Greek customs in Chicago through them. We had many interesting conversations both at the restaurant as well as at their homes. I am sure that we will keep in touch and we will have the opportunity to meet again in the future. I also had a great time with Marina and Jesus, as well as with Dimitri M and Giota B; thank you guys. Finally, I am also grateful to Ari and the time that we had the chance to spend together during my last semester in Chicago. I wish him best of luck now that he is starting his journey in the brave new world.

I was also fortunate that I had wonderful office mates all these years. With Kathy, Yun, Danko, Xiangcheng, Xudong, and Andy we shared our joys and sorrows and gave courage to each other to continue towards our goals.

I am also indebted to the rest of my friends in Chicago. To Deniz, Genady, Paul V, Ali, Tim, Mechie, Rajmonda, Anushka, Luigi, Jorge, Tuan, Matt B, and Sanja I owe at least a big thank you for being part of my life in Chicago.

On the other hand, partly because of the dedication that a PhD requires, and partly because of the time-zone difference I may have neglected some of my friends in Greece. If that was the case, I apologize. However, now that we are closer we will have the opportunity to make up some of the time that we have lost.

I have also been very fortunate to have a big and supporting family. My grandparents Nikos, Alexandra, Dimitris, and Andromachi, as well as Thodoris, Stella, Thanasis, Maria P, Grigoris, Andreas, Christina, and Anastasia have always stood next to me and supported me on my endeavors and goals. To my parents and my sister I owe everything, so this thesis is naturally dedicated to them.

Closing, there are some more people that I would like to thank.

Robert Langlois, former visiting Professor at UIC, who now works at the Frank Lab of Columbia University, had pinpointed the direction of multiple-instance active learning. His suggestion opened the path for our involvement with multiple-instance learning and active learning. I am indebted to him for proposing such an interesting direction of research.

With Balázs Szörényi I had many interesting discussions both in Chicago as well as in Hungary. I am also grateful to him when he spent his time and we went through notions that are related to Statistical Queries.

I am also indebted to my adviser for my Master's thesis [42], Professor Ioannis Emiris. During the years that I was pursuing my PhD in Chicago we have kept in touch and he has pushed me for further opportunities. One such example is [43] which is a ten page abstract of my Master's thesis. Moreover, I had the opportunity in different occasions, not only to discuss my research with him, but also present my work in the theory seminar at the Department of Informatics and Telecommunications in Athens. These invitations had many positive sides. In one occasion, I had the incentive to revise material and recall details from past work. In another occasion, one of the talks was an incentive to prepare slides for material that was still very fresh. Further, these seminars acted like a magnet and I had the chance to meet people and friends from my life during my earlier studies at the University of Athens, talk to them about my research, and listen to a wider variety of comments.

Finally, I am also indebted to my adviser for my Undergraduate thesis [41], Professor Panagiotis Stamatopoulos. Through him, his courses, as well as my thesis, I formed a broad interest in different fields of artificial intelligence early in my career. This in turn made me more confident to apply to a wider spectrum of postdoctoral positions. Hence, to a big extent, I owe him my confidence and decision to continue my work in artificial intelligence, combining agents and learning, as a postdoctoral research associate at the University of Edinburgh under the guidance of Professors Subramanian Ramamoorthy and Michael Rovatsos.

Dimitrios I. Diochnos
Edinburgh
July 2013

Contents

List of Algorithms

List of Tables

List of Figures

List of Abbreviations

AL	Active learning
MIL	Multiple-instance learning
MIAL	Multiple-instance active learning
e	Euler's constant; $2.718281828459045\ldots$
$\lg(x)$	The logarithm of x in base 2
$\ln(x)$	The natural logarithm of x
$\log(x)$	The logarithm of x in base 10
$H(x)$	The binary entropy of x

Summary

THIS thesis is concerned with problems that arise in learning theory as well as with an investigation of a popular commonsense knowledge base that is publicly available online (ConceptNet 4) with the tools of network analysis and reasoning with the knowledge that is available in the database.

In Chapter 3 we study the evolvability of monotone conjunctions under the uniform distribution through an intuitive neighborhood that was suggested by Leslie Valiant in his seminal paper that introduced evolvability. In that paper Valiant proved the evolvability of monotone conjunctions under the uniform distribution in $\mathcal{O}\left(n\lg(n/\varepsilon)\right)$ iterations using total sample size $\mathcal{O}\left((n/\varepsilon)^6\right)$. We give a structure theorem of best approximations and improve this result in $\mathcal{O}\left(\lg(1/\varepsilon) + n\lg(1/\delta)\right)$ iterations using total sample size $\tilde{\mathcal{O}}\left(n^2/\varepsilon^2 + n/\varepsilon^4\right)$, where $\tilde{\mathcal{O}}\left(\cdot\right)$ is ignoring poly-logarithmic factors. We examine the same algorithm under μ-nondegenerate product distributions and show the existence of local optima. We then switch to covariance as the fitness metric and show that a similar structure theorem for best approximations holds under μ-nondegenerate product distributions. We prove the evolvability of short monotone conjunctions under μ-nondegenerate product distributions in $\mathcal{O}\left((n/\mu)\ln(1/\varepsilon) + n\ln(1/\delta)\right)$ iterations using total sample size $\tilde{\mathcal{O}}\left(n(1/\mu)^5(1/\varepsilon)^{(4/\mu)\ln(1/\mu)}\right)$, where again $\tilde{\mathcal{O}}\left(\cdot\right)$ is ignoring poly-logarithmic factors.

In Chapter 4 we study halfspaces under the multiple instance learning (MIL) framework. Using points from the moment curve (cyclic polytopes) it is shown that the VC dimension of d-dimensional halfspaces is $\Omega(d\lg r)$ improving the previous lower bound of Sabato and Tishby of $\Omega(\lg r)$ and matching the upper bound. Using in addition Ramsey theory this result is also shown to hold over any large point set in general position. Further, it is shown that the hypothesis finding problem is NP-complete when the bags of instances are drawn from arbitrary distributions. However, the actual learning problem of d-dimensional halfspaces under the MIL setup occurs when the bags of instances are drawn from product distributions.

In Chapter 5 we examine the disagreement coefficient of monotone conjunctions under the uniform distribution. The disagreement coefficient was introduced by Hanneke in the framework of active learning. It is a combinatorial parameter that depends on the concept class, the distribution, and the actual target being learned. We study the the disagreement coefficient of monotone conjunctions under the uniform distribution and give the following results. For targets of size 0 and 1 we compute the disagreement coefficient exactly and it is $2 - 2^{1-n}$. For targets of size $2 \leqslant k \leqslant \lfloor n/2 \rfloor$ it is shown that the disagreement coefficient is $\Theta\left(2^k\right)$. For targets of size $\lfloor n/2 \rfloor + 1$ we give a lower bound of $\Omega\left(\frac{1}{n} \cdot 2^{(H(1-(\lfloor n/2 \rfloor+1)/n)-(1-(\lfloor n/2 \rfloor+1)/n))n}\right)$, where $H\left(a\right)$ is the binary entropy of a, and an upper bound of $\mathcal{O}\left(2^{\lfloor n/2 \rfloor+1}\right)$. For targets of size $\lfloor n/2 \rfloor + 2 \leqslant k \leqslant 2n/3$ we give a lower bound of $\Omega\left(\frac{1}{n} \cdot 2^{(H(1-k/n)-(1-k/n))n}\right)$ and an upper bound of $\mathcal{O}\left(2^{(H(1-k/n)-(1-k/n))n}\right)$, where again in both cases $H\left(a\right)$ is the binary entropy of a. Finally, for targets of size $k > 2n/3$ we give a lower bound of $\Omega\left(\frac{1}{n} \cdot \left(\frac{3}{2}\right)^n\right)$ and an upper bound of $\mathcal{O}\left(\left(\frac{3}{2}\right)^n\right)$.

In Chapter 6 we are investigating the commonsense knowledge base ConceptNet 4 which is publicly available online. We use the tools of network analysis and identify various properties of the induced directed and undirected multigraphs and graphs. Our findings identify missing links from the database, locate spurious links that already exist in the database, as well as provide additional knowledge to be added in the knowledge base. We also use the database for question answering. In this direction we apply variants of spreading activation techniques. Our approach gives explanations for bad results that were obtained for some questions in a previous study that applied a similar algorithm in a low-rank approximation of the adjacency matrix and improves the candidate answers for the same questions. In addition, we mine frequent rules from the database. This rule mining approach suggests some interesting possibilities. First of all, the primary aim of this rule mining approach is to add general rules that would allow further or more elaborate reasoning. Moreover, this rule mining approach can be used as an additional tool for identifying wrong assertions that are introduced in the database. Finally, we also identify rules that may make sense as factual statements about the world but not in terms of natural language usage.

Primary Subject Category: Computer Science
Secondary Subject Category: Mathematics, Artificial Intelligence
Keywords: evolvability, multiple-instance learning, active learning, knowledge bases, network analysis, Boolean functions

Chapter 1

Introduction

Computer science was arguably established formally by Turing in [135] but its roots are really old and most likely co-exist with mathematics. In contrast to other sciences, but similar to mathematics, computer science has proofs and theorems. In other words, we are not dealing with natural laws that are not known with certainty. Knuth describes computer science as *"the study of algorithms"* [78]. *Computational complexity*[1] [125, 105] captures the idea of analyzing the requirements of time and space of algorithms solving specific problems, or even at a more fundamental level whether specific problems are solvable or not. However, the practical performance of specific algorithms is also of interest even when we do not have sharp bounds on the analysis, guarantees for finding solutions and/or their quality, or more broadly, simply because we want to examine the *behavior* of an algorithm, where the 'behavior' will be a concrete notion related to a problem of interest[2]. In particular, *randomization* in algorithms tends to bring elegance and compactness into the proofs and typically translates to simple, robust, and efficient code for real-world applications. One of the main themes of this thesis will be the analysis of *randomized algorithms* [97] in a specific context which will be clear below.

Turing, in his pioneering paper [136], set the foundations for the development of machines that exhibit intelligence, thereby, introducing the field that is nowadays known as *artificial intelligence* [113, 52]. Throughout the years artificial intelligence has expanded to a large field by being inspired and absorbing results and theories from different branches of sciences such as biology, cognition, and psychology. However, reflecting our opinion and our view, artificial intelligence is to the biggest extent a sub-field of modern computer science. Towards the development of machines that act in an intelligent manner the notions of *learning* and *reasoning* are of foremost importance.

In computer science terminology the notion of 'learning' is typically further divided into *machine learning* [96] and *computational learning theory* [5, 76] although this distinction is not always very clear. Essentially, computational learning theory refers to the field that performs a rigorous analysis of algorithms used in machine learning and this is why it is considered to fall under the heading of theoretical computer science, while machine learning is typically considered to be a branch of artificial intelligence. A crucial term, which also acts as a guide for this thesis, is the notion of the *efficiency* of a learning algorithm. In this thesis, unless otherwise stated, an algorithm will be called *efficient in space* and *efficient in time* if its space and time requirements respectively are bounded from above by some polynomial expression of the input parameters[3]. Under that perspective, the foundations of computational learning theory were set in Valiant's seminal paper [137] even though theoretical

[1] *Theory of computation* is a synonym to the term 'computational complexity' in this context.

[2] For example, another interpretation of the 'behavior' of an algorithm can be its practical running time, as opposed to its theoretical worst running time. For instance, it was known for years that the simplex algorithm behaves well in practice despite its worst case exponential running time. Eventually this was justified by Spielman and Teng in their influential paper [129] which set the foundations of *smoothed analysis* of algorithms.

[3] We may simply state that an algorithm is *efficient* if it is clear from the context whether we refer to space or time requirements. Moreover, note that if an algorithm is efficient in time, then it is also efficient in space.

investigations had been made in the past that did not take efficiency into account; see e.g., [54]. The first part of this thesis will be concerned with the analysis of randomized algorithms that arise in various frameworks in the context of computational learning theory.

As far as 'reasoning' is concerned, we refer to the problem of *knowledge representation and common-sense reasoning with the represented knowledge* [11, 17]. Even though Turing's work in [136] implicitly sets the problem of reasoning with represented knowledge, McCarthy in his seminal paper [93] made this problem explicit, thereby, establishing the field of knowledge representation and common-sense reasoning. It is widely accepted that large amounts of common-sense data are required for common-sense reasoning. This in turn has resulted in the generation of common-sense knowledge bases which are also publicly available in recent years, such as Cyc [84, 83] and ConceptNet [87, 128]. Moreover, there is an increased interest in common-sense reasoning because of its potential applications to different fields, such as web search and robotics. Our focus is on ConceptNet. In particular, in this thesis we will use the tools of *network analysis* [18, 44, 99] in order to understand better the structure of such networks. Even though this exploration is interesting in its own right, our primary goal is to develop methods that are potentially useful for improving the performance of the knowledge base on various common-sense reasoning tasks. For instance, one possibility of improving the performance is the identification of missing or incorrect links in the database. Going one step beyond, our results can be considered as the first step towards the formation of a bigger collection of knowledge facts that can in turn be used as additional tools for reasoning with the aid of the databases. Finally, we also address the problem of question answering by using our implementation of a *spreading activation* [107, 25, 3] method noting that knowledge representation and reasoning are often weak spots for question answering [11, p. 780].

1.1 Instead of Contents

In Chapter 2 we present background knowledge and tools that are necessary for the results that follow in the rest of the thesis. The main tools for probabilistic analysis are different versions of the Chernoff bounds [21] (see also [58]) and the Hoeffding bound [69] in order to restrict bad events in the analysis. These tools are used in Chapter 3. We also present basic approximation and bounding techniques which are typically used in the analysis that follows. In particular we emphasize on the bounds for binomial coefficients as well as sums of binomial coefficients. These bounds on the binomial coefficients are used extensively in Chapter 5. Further, we also present the definitions of the moment curve, the cyclic polytopes, and a Ramsey theorem [57] which are used in the analysis of Chapter 4. We conclude Chapter 2 with a brief presentation of Valiant's *Probably Approximately Correct* (PAC) model of learning [137] and Sauer's lemma [116, 122, 141] from VC theory [140, 141].

Chapter 3 deals with Valiant's recently introduced framework for learning called *evolvability* [138, 139] which has already attracted the attention of people working in theoretical computer science and complexity [1, 46, 47, 48, 94, 73, 72]. It suggests a formal theory based on Darwin's theory of evolution [30]. The purpose is to allow and explain the evolution of complex mechanisms in realistic population sizes within realistic time periods. Evolution is treated as a form of computational learning from examples (experiences). Learning is influenced only by the *fitness* of the hypotheses on the examples, and not otherwise by the specific examples. This is of primary importance because of the idea that the relationship between the genotype and phenotype may be extremely complicated, and the evolutionary algorithm does not understand it. Traditional terminology would characterize evolvability as a special type of *local search*. Feldman showed in [46] that evolvability is equivalent to learning with *correlational statistical queries* [19] which is a restricted form of PAC learning [137]; see also [76, 5, 96] and Chapter 2. However, this characterization result is the product of a simulation argument that most likely does not capture how evolution is performed by nature, in the sense of intuitive algorithms. Under that perspective, evolvability is still at its infancy and our work focuses on intuitive algorithms for Boolean functions. In particular we will examine the evolvability of monotone conjunctions under the uniform distribution. We will then examine the same algorithm under μ-nondegenerate product distributions and show the existence of local optima. Finally we will switch to covariance as the fitness metric and study the same algorithm under μ-nondegenerate product distributions for short monotone conjunctions. The results of this chapter appeared in [40].

Chapter 4 deals with multiple instance learning (MIL) which is another variant of the PAC model introduced by Dietterich, Lathrop, and Lozano-Pérez in [36] who investigated the problem of drug activity prediction. In MIL the learner receives *bags* of examples instead of individual examples. The label of a bag is positive if the bag contains at least one positive example, and negative otherwise. The learning task is to infer the requirements for the observed classification of bags and predict the classification of other bags. A basic tool on the characterization of the difficulty of a problem is the VC dimension; see [141, 16] and Chapter 2. The VC dimension of d-dimensional halfspaces is $d+1$. Our work studies this problem for halfspaces in the MIL setting where we give an explicit construction for the lower bound and we also show that the same lower bound holds for halfspaces over any large point set in general position. The results rely on points from the *moment curve* (*cyclic polytopes*) [90], and the second result also uses Ramsey theory [57]. To the best of our knowledge this is the first application of cyclic polytopes in learning theory. Finally, we show that the hypothesis finding problem is NP-complete using a variant of the reduction that was used in Kundakcioglu, Seref, and Pardalos [81]. This contrasts with the polynomial time algorithm of Blum and Kalai [14] for multi-instance learning any class learnable with statistical queries, which thus applies to halfspaces as well. The fine point for this distinction is that our result implies non-learnability if the distribution on the bags can be arbitrary, while in Blum and Kalai's setting the bags of instances come from a product distribution, which is the actual learning problem. The results of this chapter appeared in [39].

Chapter 5 deals with active learning (AL) [119, 63] which is yet another variant of PAC learning developed in recent years together with the significant increase of unlabeled data for all sorts of tasks which are often available through the World Wide Web. Without loss of generality, in AL the learner queries points from a pool with unlabeled data points. The objective is to minimize the number of labels requested and form a good hypothesis. An important combinatorial parameter for the label complexity in AL is the *disagreement coefficient*, which was introduced by Hanneke in [62]. The disagreement coefficient depends on the concept class, the distribution, and the actual target being learned. However, the disagreement coefficient has so far been studied only for continuous concept classes. To the best of our knowledge, in this thesis the disagreement coefficient is studied for the first time for Boolean concept classes. More broadly, apart from a very recent paper of Balcan, Berlind, Ehrlich, and Liang [8], we are not aware of any other study of Boolean functions in the framework of active learning. In Chapter 5 we give the following results for the disagreement coefficient of monotone conjunctions under the uniform distribution. For targets of size 0 and 1 the disagreement coefficient is $2-2^{1-n}$ in every case. For targets of size $2 \leqslant k \leqslant \lfloor n/2 \rfloor$ it is shown that the disagreement coefficient is $\Theta\left(2^k\right)$. For targets of size $\lfloor n/2 \rfloor + 1$ we give a lower bound of $\Omega\left(\frac{1}{n} \cdot 2^{(H((\lceil n/2 \rceil - 1)/n) - (\lceil n/2 \rceil - 1)/n)n}\right)$, where $H(a)$ is the binary entropy of a, and an upper bound of $\mathcal{O}\left(2^{\lfloor n/2 \rfloor + 1}\right)$. For targets of size $\lfloor n/2 \rfloor + 2 \leqslant k \leqslant 2n/3$ we give a lower bound of $\Omega\left(\frac{1}{n} \cdot 2^{(H(1-k/n)-(1-k/n))n}\right)$ and an upper bound of $\mathcal{O}\left(2^{(H(1-k/n)-(1-k/n))n}\right)$, where again in both cases $H(a)$ is the binary entropy of a. Finally, for targets of size $k > 2n/3$ we give a lower bound of $\Omega\left(\frac{1}{n} \cdot \left(\frac{3}{2}\right)^n\right)$ and an upper bound of $\mathcal{O}\left(\left(\frac{3}{2}\right)^n\right)$.

Chapter 6 deals with our investigation of ConceptNet 4 that was described earlier. We perform a detailed computational study of the graphs induced by ConceptNet 4 [4] using the tools of network analysis. Moreover, our work continues the work of Ohlsson et al. [101]. We have studied various versions of the spreading activation algorithm and its relationship to algorithms running on various low-rank approximations of the matrix representing the knowledge base. Our past and ongoing work aims to address such fundamental, and so far less understood questions such as how far can one go with

[4] Homepage: `http://csc.media.mit.edu/docs/conceptnet/`.

the information contained in the knowledge bases in terms of question answering. For example, using questions on IQ tests for children, how do the statistical and logical approaches compare in terms of their question answering power, and what are useful ways to combine these approaches in the context of question answering? Spreading activation in semantic networks and link analysis techniques have been used in many contexts such as information retrieval and web search; e.g., [27, 2, 115]. Commonsense reasoning in knowledge bases appears to provide interesting new aspects for studying these and similar techniques, for example, for the identification of missing data and the correction of errors in the knowledge base. The results of this chapter appeared in [38, 12].

Chapter 2

Tools and Background

In this chapter we will see some basic notions and tools from probability theory. Some of these will be used later on in applications. Moreover, a brief introduction to PAC learning will be given together with some basic facts from VC theory.

2.1 Basics

Definition 2.1.1 (Probability Mass Function (PMF))**.** The PMF p_X of a discrete random variable X is a function that describes the *probability mass* of each (discrete) value x that X can take; that is

$$p_X(x) = \mathbf{Pr}\left(X = x\right) \, .$$

2.1.1 Discrete Random Variables

Definition 2.1.2 (Bernoulli Random Variable)**.** Let X be a Bernoulli random variable that takes two values 0 and 1 depending on the outcome of a random process (e.g. tossing a coin once). For some p with $0 \leqslant p \leqslant 1$, its PMF is

$$p_X(x) = \begin{cases} p & , \text{ if } x = 1, \\ 1 - p & , \text{ if } x = 0. \end{cases}$$

The expected value of X is $\mathbf{E}\left[X\right] = p$, while the variance is $\mathbf{Var}\left[X\right] = p(1 - p)$.

Definition 2.1.3 (Binomial Random Variable)**.** Let Y be a Binomial random variable with parameters N and p that is constructed by N Bernoulli random variables $X_1, \ldots, X_N$, each of which is 1 with probability p. It is defined as the sum $Y = \sum_{i=1}^{N} X_i$. Its PMF is

$$p_Y(k) = \mathbf{Pr}\left(Y = k\right) = \binom{N}{k} p^k (1 - p)^{N-k}, \qquad k = 0, 1, \ldots, N.$$

The expected value of Y is $\mathbf{E}\left[Y\right] = Np$, while the variance is $\mathbf{Var}\left[Y\right] = Np(1 - p)$.

Note that $\sum_{k=0}^{N} p_Y(k) = 1$.

Definition 2.1.4 (Geometric Random Variable)**.** Given a sequence of Bernoulli random variables $X_1, X_2, \ldots$, each of which is 1 with probability p, Z is a Geometric random variable expressing the minimum i such that $X_i = 1$. Its PMF is

$$p_Z(k) = (1 - p)^{k-1} p, \qquad k = 1, 2, \ldots \, .$$

The expected value of Z is $\mathbf{E}\left[Z\right] = 1/p$, while the variance is $\mathbf{Var}\left[Z\right] = \frac{1-p}{p^2}$.

Note that $\sum_{k=1}^{\infty} p_Z(k) = 1$.

Definition 2.1.5 (Poisson Random Variable)**.** Let S be a Poisson random variable with parameter $\lambda > 0$ and PMF given by

$$p_S(k) = e^{-\lambda}\frac{\lambda^k}{k!}, \qquad k = 0, 1, \ldots, N\,.$$

The expected value of S is $\mathbf{E}[S] = \lambda$, and the variance is also $\mathbf{Var}[S] = \lambda$.

Note that $\sum_{k=0}^{N} p_S(k) = 1$.

2.1.2 Bernoulli Process

Informally it is a sequence of independent coin tosses.

Definition 2.1.6 (Bernoulli process)**.** It is a sequence $X_1, X_2, \ldots$ of independent Bernoulli random variables X_i such that for every i it holds:

$$\begin{cases} \mathbf{Pr}(X_i = 1) & = & \mathbf{Pr}(\text{success at the } i\text{th trial}) & = & p \\ \mathbf{Pr}(X_i = 0) & = & \mathbf{Pr}(\text{failure at the } i\text{th trial}) & = & 1 - p \end{cases}$$

2.2 Approximating and Bounding

Basic tools on approximating and bounding quantities are presented below.

2.2.1 The Cauchy-Schwartz Inequality

$$\left(\sum_{i=1}^{n} x_i y_i\right)^2 \leqslant \left(\sum_{i=1}^{n} x_i^2\right)\left(\sum_{i=1}^{n} y_i^2\right) \tag{2.1}$$

2.2.2 Bounding Combinations

Let $1 < k < n$, with $k, n \in \mathbb{N}$. Then,

$$\left(\frac{n}{k}\right)^k < \binom{n}{k} < \left(\frac{n \cdot e}{k}\right)^k \qquad \text{and} \qquad e \cdot \left(\frac{n}{e}\right)^n < n! < e \cdot \left(\frac{n+1}{e}\right)^{n+1} \tag{2.2}$$

One can also bound a binomial coefficient with the help of the binary entropy function as shown below.

Proposition 2.2.1 (Bounding a Binomial Coefficient with the Binary Entropy)**.** *Let* $0 < k < n$*, with* $k, n \in \mathbb{N}$*. Moreover, let* $k/n = \alpha \in (0, 1)$*. Then,*

$$\frac{2^{H(\alpha)\cdot n}}{n+1} < \binom{n}{k} = \binom{n}{\alpha \cdot n} < 2^{H(\alpha)\cdot n},$$

where $H(a)$ *is the binary entropy of* a*; that is,* $H(a) = -a \cdot \lg a - (1 - a) \cdot \lg(1 - a)$ *and* $\lg a$ *is the logarithm of* a *in base 2.*

Proof. We examine each inequality separately.
Lower Bound. Let $f(k) = \binom{n}{k} \cdot \alpha^k \cdot (1 - \alpha)^{n-k}$. We have

$$\begin{aligned} f(k+1) - f(k) &= \binom{n}{k+1} \cdot \alpha^{k+1} \cdot (1-\alpha)^{n-k-1} - \binom{n}{k} \cdot \alpha^k \cdot (1-\alpha)^{n-k} \\ &= \binom{n}{k} \cdot \alpha^k \cdot (1-\alpha)^{n-k} \cdot \left(\frac{(n-k)}{(k+1)} \cdot \frac{\alpha}{(1-\alpha)} - 1\right) \end{aligned}$$

Note that the denominator is positive since $\alpha \in (0,1)$. Hence we have $f(k+1) - f(k) > 0 \iff (n-k) \cdot \alpha - (k+1) \cdot (1-\alpha) > 0 \iff \alpha \cdot n - \alpha \cdot k - k + \alpha \cdot k - 1 + \alpha > 0 \iff k < \alpha \cdot n - (1-\alpha)$. Again we note that $\alpha \in (0,1)$ and hence $0 < 1-\alpha < 1$. Since k is an integer, it follows that $f(k+1) > f(k)$ for every $k < \alpha \cdot n$. Moreover, with the same argument we have that $f(k+1) < f(k)$ for every $k \geqslant \alpha \cdot n$. As a consequence, the function $f(k)$ attains its maximum for $k = \alpha \cdot n$. We will now use this fact. By the binomial theorem we have $1 = (\alpha+(1-\alpha))^n = \sum_{i=0}^{n} \binom{n}{i} \cdot \alpha^i \cdot (1-\alpha)^{n-i} < (n+1) \cdot \binom{n}{\alpha \cdot n} \cdot \alpha^{\alpha \cdot n} \cdot (1-\alpha)^{n-\alpha \cdot n}$. In other words

$$\binom{n}{\alpha \cdot n} > \frac{\alpha^{-\alpha \cdot n} \cdot (1-\alpha)^{-(1-\alpha)n}}{(n+1)} = \frac{2^{H(\alpha) \cdot n}}{n+1} .$$

Upper Bound. Again from the binomial theorem we have $1 = (\alpha + (1-\alpha))^n = \sum_{i=0}^{n} \binom{n}{i} \cdot \alpha^i \cdot (1-\alpha)^{n-i} > \binom{n}{\alpha \cdot n} \cdot \alpha^{\alpha \cdot n} \cdot (1-\alpha)^{n-\alpha \cdot n}$. In other words

$$\binom{n}{\alpha \cdot n} < \alpha^{-\alpha \cdot n} \cdot (1-\alpha)^{-(1-\alpha) \cdot n} = 2^{H(\alpha) \cdot n} .$$

The proposition follows by combining the above two cases. □

In fact, as we will see below, one can do better and give the same upper bound for the entire summation of the binomial coefficients up to $k = \lfloor \alpha \cdot n \rfloor$, where $\alpha \in (0, 1/2]$.

Proposition 2.2.2 (Upper Bound on the Sum of Binomial Coefficients with the Binary Entropy)**.** *Let $n \geqslant 1$ and $0 < \alpha \leqslant 1/2$. Then,*

$$\sum_{k=0}^{\lfloor \alpha \cdot n \rfloor} \binom{n}{k} < 2^{H(\alpha) \cdot n} ,$$

where $H(a)$ is the binary entropy of a; that is, $H(a) = -a \cdot \lg a - (1-a) \cdot \lg(1-a)$ and $\lg a$ is the logarithm of a in base 2.

Proof. Before we proceed with the actual proof we note that for $\alpha \in (0, 1/2]$ it holds $\lg(\alpha) - \lg(1-\alpha) \leqslant 0$. This follows immediately since the function $f(x) = \lg(\alpha) - \lg(1-\alpha)$ for $\alpha \in (0, 1/2]$ is monotone increasing and $f(1/2) = 0$. Now let $i \in [0, \lfloor \alpha \cdot n \rfloor]$ and hence the quantity $(\alpha \cdot n - i)$ is non-negative. Then, we have

$$\begin{aligned} (\alpha \cdot n - i) \cdot \lg(\alpha) - (\alpha \cdot n - i) \cdot \lg(1-\alpha) &\leqslant 0 \\ -\alpha \cdot n \cdot \lg(\alpha) + i \cdot \lg(\alpha) + \alpha \cdot n \cdot \lg(1-\alpha) - i \cdot \lg(1-\alpha) &\geqslant 0 \\ i \cdot \lg(\alpha) - i \cdot \lg(1-\alpha) &\geqslant \alpha \cdot n \cdot \lg(\alpha) - \alpha \cdot n \cdot \lg(1-\alpha) . \end{aligned}$$

Adding $n \cdot \lg(1-\alpha)$ on both sides we get

$$i \cdot \lg(\alpha) + (n-i) \cdot \lg(1-\alpha) \geqslant n \cdot \alpha \cdot \lg(\alpha) + n \cdot (1-\alpha) \cdot \lg(1-\alpha) ,$$

where by using the definition of the entropy function we have

$$\alpha^i \cdot (1-\alpha)^{n-i} \geqslant 2^{-n \cdot H(\alpha)} . \tag{2.3}$$

We are now ready to proceed with the actual proof of the statement. Using the binomial theorem we have

$$\begin{aligned}
1 = (\alpha + (1-\alpha))^n &= \sum_{i=0}^{n} \binom{n}{i} \cdot \alpha^i \cdot (1-\alpha)^{n-i} \\
&> \sum_{i=0}^{\lfloor \alpha \cdot n \rfloor} \binom{n}{i} \cdot \alpha^i \cdot (1-\alpha)^{n-i} \\
&\geqslant \sum_{i=0}^{\lfloor \alpha \cdot n \rfloor} \binom{n}{i} \cdot 2^{-n \cdot H(\alpha)} \\
&= 2^{-n \cdot H(\alpha)} \cdot \sum_{i=0}^{\lfloor \alpha \cdot n \rfloor} \binom{n}{i},
\end{aligned}$$

where in the third line we used (2.3). The proposition follows. □

Finally, another useful bound for the sum of the binomial coefficients is given below.

Proposition 2.2.3 (General Upper Bound on the Sum of Binomial Coefficients). *Let* $0 < d < n$, *with* $d, n \in \mathbb{N}$. *Then,*

$$\sum_{k=0}^{d} \binom{n}{k} < \left(\frac{e \cdot n}{d}\right)^d .$$

Proof. Since $0 < d/n < 1$ we may write

$$\left(\frac{d}{n}\right)^d \cdot \sum_{i=0}^{d} \binom{n}{i} < \sum_{i=0}^{d} \left[\binom{n}{i} \cdot \left(\frac{d}{n}\right)^i\right] < \sum_{i=0}^{n} \left[\binom{n}{i} \cdot \left(\frac{d}{n}\right)^i\right] = (1 + d/n)^n \leqslant e^d .$$

The proposition follows. □

2.2.3 Common Approximations

Very often in the analysis we want to bound expressions of the form $(1-x)^n$ from above, with $x \in (0,1)$; typically x will be a probability of a good event happening. In such cases we will use the inequality

$$(1-x)^n \leqslant e^{-x \cdot n} \tag{2.4}$$

without any further justifications. Note that (2.4) is valid, since for every $x \in \mathbb{R}$ it holds $1 + x \leqslant e^x$. We also note here that for $x \in [0, 1/2]$ it holds

$$1 - x \geqslant e^{-2x} . \tag{2.5}$$

Proposition 2.2.4 (Poisson Approximation). *The Poisson PMF with parameter* λ *is a* good *approximation for a binomial PMF with parameters* N *and* p, *provided that* $\lambda = Np$, N *is very large, and* p *is very small.*

2.2.4 Bounding Probabilities

Proposition 2.2.5 (Union Bound). *Let* $Y_1, Y_2, \ldots, Y_S$ *be* S *events in a probability space. Then* $\mathbf{Pr}\left(\bigcup_{j=1}^{S} Y_j\right) \leqslant \sum_{j=1}^{S} \mathbf{Pr}\left(Y_j\right)$. *The inequality is equality for disjoint events* Y_j.

Proposition 2.2.6 (Markov's Inequality). *Any non-negative random variable* X *satisfies*

$$\mathbf{Pr}\left(X \geqslant \alpha\right) \leqslant \frac{\mathbf{E}\left[X\right]}{\alpha}, \qquad \forall \alpha > 0\,.$$

Proposition 2.2.7 (Chebyshev's Inequality). *Let* X *be a random variable with expected value* μ *and variance* σ^2. *Then*

$$\mathbf{Pr}\left(|X - \mu| \geqslant \alpha\right) \leqslant \frac{\sigma^2}{\alpha^2}, \qquad \forall \alpha > 0\,.$$

Remark 2.2.8 (Chebyshev vs. Markov). The Chebyshev inequality tends to give better bounds than the Markov inequality, because it also uses information on the variance of X.

Theorem 2.2.9 (Weak Law of Large Numbers). *Let* $X_1, \ldots, X_N$ *be a sequence of* independent identically distributed *random variables, with expected value* μ. *For every* $\epsilon > 0$:

$$\mathbf{Pr}\left(\left|\frac{1}{N}\sum_{i=1}^{N} X_i - \mu\right| \geqslant \epsilon\right) \to 0, \qquad \textit{as} \;\; N \to \infty \tag{2.6}$$

Proof. Let $X_1, \ldots, X_N$ be a sequence of *independent identically distributed* random variables, with expected value μ and variance σ^2. Define the random variable $Y = \frac{1}{N}\sum_{i=1}^{N} X_i$. By linearity of expectation we get $\mathbf{E}\left[Y\right] = \frac{1}{N}\sum_{i=1}^{N} \mathbf{E}\left[X_i\right] = \mu$. Since all the X_i are independent, the variance is $\mathbf{Var}\left[Y\right] = \frac{1}{N^2}\sum_{i=1}^{N} \mathbf{Var}\left[X_i\right] = \frac{\sigma^2}{N}$. We now apply Chebyshev's inequality and obtain $\mathbf{Pr}\left(|Y - \mu| \geqslant \epsilon\right) \leqslant \frac{\sigma^2}{N\epsilon^2}$, for any $\epsilon > 0$. □

Concentration and Tail Inequalities

In this section we examine a series of tools for estimating the concentration and bounding the probability of the tails.

Proposition 2.2.10 (Hoeffding Bound [69, 35]). *Let* $X_1, \ldots, X_R$ *be* R *independent random variables, each taking values in the range* $\mathfrak{I} = [\alpha, \beta]$. *Let* μ *denote the mean of their expectations. Then*

$$\mathbf{Pr}\left(\left|\frac{1}{R}\sum_{i=1}^{R} X_i - \mu\right| \geqslant \epsilon\right) \leqslant e^{-2R\epsilon^2/(\beta-\alpha)^2}\,.$$

Proposition 2.2.11 (Chernoff Bound for Upper Tail). *Assume* $X_1, X_2, \ldots, X_t$ *are independent Poisson trials. Let* $X = \sum_{i=1}^{t} X_i$, *and* $\mu = \mathbf{E}\left[X\right]$. *Then, for* $\gamma \in (0, 1)$ *it holds*

$$\mathbf{Pr}\left(X > (1+\gamma)\mu\right) \leqslant e^{-\mu\gamma^2/3}\,.$$

Proposition 2.2.12 (General Chernoff Bound for Upper Tail). *Assume* $X_1, X_2, \ldots, X_t$ *are independent Poisson trials. Let* $X = \sum_{i=1}^{t} X_i$, *and* $\mu = \mathbf{E}\left[X\right]$. *Then, for* $\gamma \geqslant 0$ *it holds*

$$\mathbf{Pr}\left(X > (1+\gamma)\mu\right) \leqslant e^{-\mu\gamma^2/(2+\gamma)}\,.$$

Proposition 2.2.13 (Chernoff Bound for Lower Tail). *Assume* $X_1, X_2, \ldots, X_t$ *are independent Poisson trials. Let* $X = \sum_{i=1}^{t} X_i$, *and* $\mu = \mathbf{E}\left[X\right]$. *Then, for* $\gamma \in (0, 1)$ *it holds*

$$\mathbf{Pr}\left(X < (1-\gamma)\mu\right) \leqslant e^{-\mu\gamma^2/2}\,.$$

Proposition 2.2.14 (General Chernoff Bound for Lower Tail). *Assume* $X_1, X_2, \ldots, X_t$ *are independent Poisson trials. Let* $X = \sum_{i=1}^{t} X_i$, *and* $\mu = \mathbf{E}\left[X\right]$. *Then, for* $\gamma \geqslant 0$ *it holds*

$$\mathbf{Pr}\left(X < (1-\gamma)\mu\right) \leqslant e^{-\mu\gamma^2/(2+\gamma)}\,.$$

The following lemma will prove to be useful.

Lemma 2.2.15. *Tossing a biased coin that gives* H *with probability* p *for* $t = \left\lceil \frac{2}{p}\left(\kappa + \ln\left(\frac{1}{\delta_C}\right)\right)\right\rceil$ *times guarantees at least* κ H *with probability at least* $1 - \delta_C$.

Proof. Let X_i be the indicator random variable that is 1 if we observe H in the i-th coin-toss, and 0 otherwise. We have $\mu = tp$, and we set $\gamma = 1 - \kappa/\mu = 1 - \kappa/(tp)$. By Proposition 2.2.13 we have $\mathbf{Pr}(X < \kappa) \leqslant e^{-(tp-\kappa)^2/(2tp)}$. We now require to bound this quantity from above by δ_C, and want to solve for t; i.e. we want to satisfy $t^2 - \frac{2}{p} \cdot \left(\kappa + \ln\left(\frac{1}{\delta_C}\right)\right) \cdot t + \left(\frac{\kappa}{p}\right)^2 \geqslant 0$. Regarding the positive root of the last equation we have

$$\begin{aligned} t &= \frac{\frac{2}{p}\cdot\left(\kappa + \ln\left(\frac{1}{\delta_C}\right)\right) + \sqrt{\left(\frac{2}{p}\cdot\left(\kappa + \ln\left(\frac{1}{\delta_C}\right)\right)\right)^2 - 4\left(\frac{\kappa}{p}\right)^2}}{2} \\ &\leqslant \frac{\frac{2}{p}\cdot\left(\kappa + \ln\left(\frac{1}{\delta_C}\right)\right) + \sqrt{\left(\frac{2}{p}\cdot\left(\kappa + \ln\left(\frac{1}{\delta_C}\right)\right)\right)^2}}{2} \\ &= \frac{2}{p}\cdot\left(\kappa + \ln\left(\frac{1}{\delta_C}\right)\right) . \end{aligned}$$

The lemma follows. □

2.3 Polyhedral Combinatorics

A simplex in $\mathbb{R}^d$ is the convex hull of an affinely independent point-set. A face of a convex polytope P is defined as either P itself, or a asubset of P of the form $P \cap H$, where H is the hyperplane such that P is fully contained in one of the closed halfspaces determined by H. For a d-dimensional polytope P, we call 0-faces as vertices, 1-faces as edges, $(d-2)$-faces as ridges, and $(d-1)$-faces as facets.

Definition 2.3.1 (Moment Curve). The curve $\gamma = \{(t, t^2, \dots, t^d) \ : \ t \in \mathbb{R}\}$ in $\mathbb{R}^d$ is called the d-dimensional moment curve.

Lemma 2.3.2. *Any hyperplane* H *intersects the moment curve* γ *in at most* d *points. If there are* d *intersections, then* H *can not be tangent to* γ*, and thus at each intersection,* γ *passes from one side of* H *to the other.*

Definition 2.3.3 (Cyclic Polytope). The convex hull of points $x(t_1), \dots, x(t_n)$ on the d-dimensional moment curve, for $t_1 < \dots < t_n$, with $n \geqslant d+1$, is called a *cyclic polytope.*

For any $I \subseteq [n], |I| = k \leqslant \lfloor \frac{d}{2} \rfloor$, the polynomial

$$\prod_{i \in I}(u - t_i)^2 = \sum_{j=0}^{2k} a_j u^j$$

is 0 at every $t_i, i \in I$ and positive at every $t_i, i \notin I$. Thus the halfspace $-\sum_{j=1}^{2k} a_j u_j \geqslant a_0$ contains every point $x(t_i), i \in I$, and none of the points $x(t_i), i \notin I$. Thus every set of at most $\lfloor \frac{d}{2} \rfloor$ vertices forms a face of a cyclic polytope.

Proposition 2.3.4 (Gale's Evenness Criterion). *Let* V *be a vertex set of a cyclic polytope* P *considered with the linear ordering* $\leqslant$ *along the moment curve (larger vertices have larger values of the parameter* t*). Let* $F = \{u_1, u_2, \dots, u_d\} \subseteq V$ *be a* d*-tuple of vertices of* P*, where* $u_1 < u_2 < \cdots < u_d$*. Then* F *determines a facet of* P *if and only if for any two vertices* $u, v \in V \setminus F$*, the number of vertices* $v_i \in F$ *with* $u < v_i < v$ *is even.*

The facets (i.e., $(d-1)$-dimensional faces) of cyclic polytopes are described by *Gale's evenness condition*: for $t_{i_1} < \cdots < t_{i_d}$ the vertices $x(t_{i_1}), \cdots, x(t_{i_d})$ form a facet if and only if for any two other vertices $x(t_u)$ and $x(t_v)$ there are an even number of values t_{i_j} between t_u and t_v. This is proven by considering the hyperplane $\sum_{j=1}^{d} a_j w^j = -a_0$ defined by

$$\prod_{j=1}^{d}(w - t_{i_j}) = \sum_{j=0}^{d} a_j w^j.$$

The condition follows by counting the number of sign changes between t_u and t_v.

2.4 Ramsey Theory

We will need the following Proposition.

Proposition 2.4.1 ([57]). *There is a function $R(u,v)$ such that if the u-subsets of a set of size at least $R(u,v)$ are two-colored then there is a subset of size v with all its u-subsets colored the same.*

2.5 Learning Theory

Below we give a brief description of the basic notions of the Probably Approximately Correct (PAC) model of learning due to Valiant [137] (see also [5, 76, 96]) as well as the VC dimension and Sauer's lemma. These notions will be useful in later chapters.

2.5.1 The Probably Approximately Correct (PAC) Learning Model

Let X be a set called the *instance (input) space* which is the set of encodings of instances in the learner's world. Typically we will subscript with $_n$ to indicate the dimension; that is, we write write X_n. A *concept* c over X is just a subset c $\subseteq$ X. A *target* concept is the concept that the learner wants to learn. In particular, c : $X_n \rightarrow \{0, 1\}$ (or c : $X_n \rightarrow \{\text{NEGATIVE}, \text{POSITIVE}\}$), with $c(x) = 1$ (or POSITIVE) indicating that x is a positive example of c, and $c(x) = 0$ (or NEGATIVE) indicating that x is a negative example of c. A concept *class* $\mathcal{C}$ over X is a collection of concepts over X. We will denote with $\mathcal{D}_n$ the *(target) distribution* over X_n. A hypothesis concept h is a *guess* or an *approximation* of c. A hypothesis concept class $\mathcal{H}$ (over X_n) is a collection of h over X_n. If $\mathcal{H} = \mathcal{C}$ we call the setting *proper* learning. The *error* of h with respect to c is expressed by:

$$\text{error}\,(\text{h}) = \Pr_{x \in \mathcal{D}_n}\left[\text{c}(x) \neq \text{h}(x)\right].$$

In a Venn diagram (over X_n) this represents the region c $\triangle$ h; see Figure 2.1. Typically the learner is given an *accuracy* $0 < \varepsilon \leqslant 1$ and the goal is to come up with a hypothesis such that the error of the hypothesis is less than the accuracy; that is, error (h) $\leqslant \varepsilon$. Finally, we also have the notion of *confidence* δ with $0 < \delta \leqslant 1$ which expresses the probability of *failure* that the learner will come up with a hypothesis that has small error. The goal of the learner in PAC learning is to satisfy the equation

$$\mathbf{Pr}\,(\text{error}\,(\text{h}) \leqslant \varepsilon) \geqslant 1 - \delta\,. \tag{2.7}$$

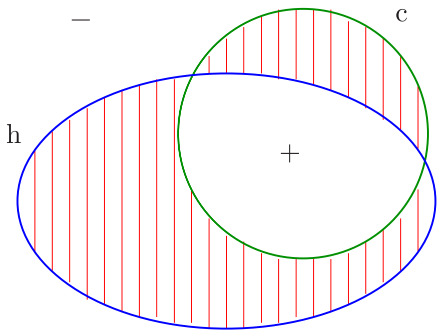


Figure 2.1: A comparison between hypothesis h and the target concept c. The shaded region indicates the error region; i.e. where h and c *disagree*.

2.5.2 VC Dimension

We say that a training sample S of length m is *shattered* by the hypothesis class $\mathcal{H}$, or that $\mathcal{H}$ *shatters* S, if all 2^m possible classifications of S can be accomplished with hypotheses in $\mathcal{H}$[1]. A hypothesis class $\mathcal{H}$ has *Vapnik-Chervonenkis dimension* (or *VC dimension*) d, denoted as $\text{VCdim}(\mathcal{H}) = d$, if there is a training sample of length d shattered by $\mathcal{H}$, and there exists no training sample of size at least $d+1$ shattered by $\mathcal{H}$. If there is no such maximum, we say that the VC dimension is *infinite*.

The growth function is defined by $\Pi_{\mathcal{H}}(m) = \max\{\Pi_{\mathcal{H}}(x) \ : \ x \in X^m\}$.

Lemma 2.5.1 (Sauer's Lemma [116, 122, 141, 5, 76]). *Let $d \geqslant 0$ and $m \geqslant 1$ be given integers and let $\mathcal{H}$ be a hypothesis space with VCdim $(\mathcal{H}) = d$. Then*

$$\Pi_{\mathcal{H}}(m) \leqslant 1 + \binom{m}{1} + \binom{m}{2} + \cdots + \binom{m}{d} = \Phi(d, m).$$

Proposition 2.5.2 ([5, 76]). *For all $m \geqslant d \geqslant 1$,*

$$\Phi(d, m) < \left(\frac{em}{d}\right)^d ,$$

where e is the base of the natural logarithms.

Theorem 2.5.3 (Finite VC Dimension and Sample Size; [5, 76]). *Let $\mathcal{C}$ be a concept class, and $\mathcal{H}$ a representation class of VC dimension $d \geqslant 1$. Then any learning algorithm that takes as input*

$$m \geqslant \left\lceil \frac{4}{\varepsilon} \cdot \left(d \cdot \ln \frac{12}{\varepsilon} + \ln \frac{2}{\delta} \right) \right\rceil$$

labeled examples of a concept in $\mathcal{C}$, and produces as output a concept $h \in \mathcal{H}$ that is consistent with the m examples is guaranteed to have small error with high probability.

Theorem 2.5.4 (Lower Bound; [45]). *Assume $0 < \varepsilon \leqslant 1/8$ and $0 < \delta \leqslant 1/100$. Let $\mathcal{C}$ be a concept class with VC dimension $d \geqslant 2$. Then, any PAC learning algorithm for $\mathcal{C}$ must use sample size*

$$m \geqslant \frac{d-1}{32 \cdot \varepsilon} = \Omega\left(d/\varepsilon\right) .$$

[1] Clearly, we are talking about distinct samples, otherwise no $\mathcal{H}$ can shatter S.

Chapter 3

Evolvability

A model relating evolution to learning was introduced by Valiant [138] in 2007. It assumes that some functionality is evolving over time. The process of evolution is modelled by updating the representation of the current hypothesis, based on its performance for training examples. Performance is measured by the correlation of the hypothesis and the target. Updating is done using a randomized local search in a neighborhood of the current representation. The objective is to evolve a hypothesis with close to optimal performance.

As a paradigmatic example, Valiant [138] showed that monotone conjunctions of Boolean variables with the uniform probability distribution over the training examples are evolvable. Monotone conjunctions are a basic concept class for learning theory, which have been studied from several different aspects [74, 76, 110]. Valiant's algorithm, which is referred to as the *swapping* algorithm in this chapter, considers *mutations* obtained by swapping a variable for another one, and adding and deleting a variable[1], and chooses randomly among beneficial mutations (or among neutral ones if there are no beneficial mutations).

Valiant also established a connection between the model and learning with *statistical queries* (Kearns [74], see also [76]), and studied different versions such as evolution with and without initialization. Valiant noted that concept learning problems have been studied before in the framework of genetic and evolutionary algorithms (e.g., Ros [111]), but his model is different (more restrictive) in its emphasis on fitness functions which depend on the training examples *only* through their performance, and *not* on the training instances themselves. His model excludes, e.g., looking at which bits are on or off in the training examples.

Feldman [46, 47] gave general results on the model and its different variants, focusing on the relationship to statistical queries. In fact Feldman showed equivalence between evolvability and correlational statistical queries. The translation, as noted by Feldman, does not lead to the most efficient or natural[2] evolutionary algorithms in general. This is the case with monotone conjunctions: even though their evolvability follows from Feldman's result, it is still of interest to find simple and efficient evolution procedures for this class. Michael [94] showed that decision lists are evolvable under the uniform distribution using the Fourier representation.

In general, exploring the performance of *simple* evolutionary algorithms is an interesting direction of research; hopefully, leading to new design and analysis techniques for efficient evolution algorithms. The swapping algorithm, in particular, appears to be a basic evolutionary procedure (mutating features in and out of the current hypothesis) and it exhibits interesting behavior. Thus its performance over distributions other than uniform deserves more detailed study.

In this chapter we continue the study of the swapping algorithm for evolving monotone conjunctions.

[1] These mutations may be viewed as swapping a variable with the constant 1.

[2] Of course, we do not use the term 'natural' here to suggest any actual connection with evolutionary processes in nature.

A modified presentation of the algorithm for the uniform distribution is given, leading to a simplified analysis and an improved complexity bound (Theorem 3.3.8). We give a simple characterization of best approximations by short hypotheses, which is implicit in the analysis of the algorithm.

We then consider the swapping algorithm for *product distributions*. Product distributions generalize the uniform distribution, and they are studied in learning theory in the context of extending learnability results from the uniform distribution, usually under non-degeneracy conditions (see, e.g. [51, 59, 71, 118]). We show that the characterization of best approximations does not hold for product distributions in general, and that the fitness function may have local optima.

It is shown that the picture changes if we replace the correlation fitness function with *covariance*. (Using fitness functions other than correlation has also been considered by Feldman [47] and Michael [94]; the fitness functions discussed in those papers are different from covariance.) In this case there is a characterization of best approximations similar to the uniform distribution with correlation. This leads to two positive results for the evolvability of monotone conjunctions under product distributions.

Theorem 3.5.8 shows that in the *unbounded-precision* model of evolution, the swapping algorithm using covariance as the fitness function, is an efficient algorithm for monotone conjunctions over *arbitrary* product distributions. Thus this result applies to a very simply defined (though clearly not realistic) evolution model, and analyzes a very simple and natural evolution algorithm (swaps using covariance) over a whole class of distributions (product distributions without any restrictions). Therefore, it may be of interest as an initial example motivating further models and algorithms, such as the introduction of short and long hypotheses in order to work with polynomial sample size estimates of performances. Theorem 3.5.9 shows that the swapping algorithm works if the target is short and the underlying product distribution is μ-nondegenerate.

The rest of this chapter is structured as follows. Section 3.1 has an informal description of the swapping algorithm. As we are focusing on a single algorithm and its variants, we do not need to define the evolution model in general. The description of the swapping algorithm and some additional material given in Section 3.2 contain the details of the model that are necessary for the rest of this chapter. Section 3.3 contains the analysis of the swapping algorithm for the case of uniform distribution. The performance of the swapping algorithm for product distributions is discussed in Section 3.4. In Section 3.5 we turn to the swapping algorithm using covariance as fitness function. Finally, Section 3.6 contains some further remarks and open problems.

3.1 An Informal Description of the Swapping Algorithm

Given a set of Boolean variables $x_1, \ldots, x_n$, we assume that there is an unknown *target* c, a monotone conjunction of some of these variables. The possible *hypotheses* h are of the same class. The truth values TRUE and FALSE are represented by 1 and -1. The *performance* of a hypothesis h is

$$\mathrm{Perf}_{\mathcal{U}_n}(\mathrm{h}, \mathrm{c}) = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} \mathrm{h}(x) \cdot \mathrm{c}(x), \tag{3.1}$$

called the *correlation* of h and c. Here $\mathcal{U}_n$ denotes the uniform distribution over $\{0,1\}^n$. The evolution process starts with an initial hypothesis h_0, and produces a sequence of hypotheses using a random walk type procedure on the set of monotone conjunctions.

Each hypothesis h is assigned a *fitness value*, called the *performance* of h. The walk is performed by picking randomly a hypothesis h′ from the *neighborhood* of the current hypothesis h which seems to be more fit (beneficial) compared to h, or is about as fit (neutral) as h. Details are given in Section 3.2.

Some care is needed in the specification of the probability distribution over beneficial and neutral hypotheses. Moreover, there is a distinction between *short* and *long* conjunctions, and the neighborhoods they induce. Valiant uses a threshold value $q = \mathcal{O}(\log(n/\varepsilon))$ for this distinction. Section 3.3.2 has details.

Valiant showed that if this algorithm runs for $O(n\log(n/\varepsilon))$ stages, and evaluates performances using total sample size $O((n/\varepsilon)^6)$ and different tolerances for short, resp. long conjunctions, then with probability at least $1-\varepsilon$ it finds a hypothesis h with $\mathrm{Perf}_{\mathcal{U}_n}(h,c) \geqslant 1-\varepsilon$.

3.2 Preliminaries

The neighborhood N of a conjunction h is the set of conjunctions that arise by *adding* a variable, *removing* a variable, or *swapping* a variable with another one, plus the conjunction itself[3]. The conjunctions that arise by adding a variable form the neighborhood N^+, the conjunctions that arise by dropping a variable form the neighborhood N^-, and the conjunctions that arise by swapping a variable form the neighborhood N^{+-}. In other words we have $N = N^- \cup N^+ \cup N^{+-} \cup \{h\}$. As an example, let our current hypothesis be $h = x_1 \wedge x_2$, and $n = 3$. Then, $N^- = \{x_1, x_2\}$, $N^+ = \{x_1 \wedge x_2 \wedge x_3\}$, and $N^{+-} = \{x_3 \wedge x_2, x_1 \wedge x_3\}$. Note that $|N| = O(n^2)$ in general.

Lemma 3.2.1 (General Neighborhood). *The size of the neighborhood for a hypothesis of size* $|h| = k$ *is* $|N| = (k+1)\cdot n + 1 - k^2$.

Proof. There are $(n-k)$ ways to add a variable, k ways to remove a variable, and $k\cdot(n-k)$ ways to swap a variable in the current hypothesis. □

Similarity between two conjunctions h and c in an underlying distribution $\mathcal{D}_n$ is measured by the *performance* function[4] $\mathrm{Perf}_{\mathcal{D}_n}(h,c)$ which is evaluated approximately, by drawing a random sample S and computing $\frac{1}{|S|}\sum_{x\in S} h(x)\cdot c(x)$. The goal of the evolution process is to *evolve* a hypothesis h such that:

$$\mathbf{Pr}\left(\mathrm{Perf}_{\mathcal{D}_n}(h,c) < \mathrm{Perf}_{\mathcal{D}_n}(c,c) - \varepsilon\right) < \delta. \tag{3.2}$$

The accuracy parameter ε and the confidence δ are treated as one in [138].

Given a target c, we split the neighborhood in 3 parts by the *increase* in performance that they offer. There are *beneficial*, *neutral*, and *deleterious* mutations. In particular, for a given neighborhood N and real constant t (*tolerance*) we are interested in the sets

$$\begin{cases} \mathrm{Bene} &= N \cap \left\{h' \mid \mathrm{Perf}_{\mathcal{D}_n}(h',c) \geqslant \mathrm{Perf}_{\mathcal{D}_n}(h,c) + t\right\} \\ \mathrm{Neut} &= N \cap \left\{h' \mid \mathrm{Perf}_{\mathcal{D}_n}(h',c) \geqslant \mathrm{Perf}_{\mathcal{D}_n}(h,c) - t\right\} \setminus \mathrm{Bene}. \end{cases} \tag{3.3}$$

A mutation is *deleterious* if it is neither beneficial nor neutral.

The size (or length) |h| of a conjunction h is the number of variables it contains. Given a target conjunction c and a size q, we will be interested in the best size q approximation of c.

Definition 3.2.2 (Best q-Approximation). A hypothesis h is called a best q-approximation of c if $|h| \leqslant q$ and $\forall h' \neq h, |h'| \leqslant q : \mathrm{Perf}_{\mathcal{D}_n}(h',c) \leqslant \mathrm{Perf}_{\mathcal{D}_n}(h,c)$.

Note that the best approximation is not necessarily unique.

In this chapter the following performance functions are considered; the first one is used in [138] and the second one is the *covariance* of h and c[5]:

$$\mathrm{Perf}_{\mathcal{D}_n}(h,c) = \sum_{x\in\{0,1\}^n} h(x)c(x)\mathcal{D}_n(x) = \mathbf{E}[h\cdot c] = 1 - 2\cdot\mathbf{Pr}(h \neq c) \tag{3.4}$$

$$\mathbf{Cov}[h,c] = \mathrm{Perf}_{\mathcal{D}_n}(h,c) - \mathbf{E}[h]\cdot\mathbf{E}[c]. \tag{3.5}$$

[3] As h will be clear from the context, we write N instead of $N(h)$.

[4] See the end of this section for the specific performance functions considered in this chapter. For simplicity, we keep the notation Perf for a specific performance function.

[5] A related performance function, not considered here, is the *correlation coefficient*.

3.3 Monotone Conjunctions under the Uniform Distribution

Given a target conjunction c and a hypothesis conjunction h, the performance of h with respect to c can be found by counting truth assignments. Let

$$h = \bigwedge_{i=1}^{m} x_i \wedge \bigwedge_{\ell=1}^{r} y_\ell \quad \text{and} \quad c = \bigwedge_{i=1}^{m} x_i \wedge \bigwedge_{k=1}^{u} w_k. \tag{3.6}$$

Thus the x's are *mutual* variables, the y's are *redundant* variables in h, and the w's are *undiscovered*, or *missing* variables in c. Variables in the target c are called *good*, and variables not in the target c are called *bad*.

The probability of the error region is $(2^r + 2^u - 2)2^{-m-r-u}$ and so

$$\mathrm{Perf}_{\mathcal{U}_n}(h, c) = 1 - 2^{1-m-u} - 2^{1-m-r} + 2^{2-m-r-u}. \tag{3.7}$$

For a fixed threshold value q, a conjunction h is *short* (resp., *long*), if $|h| \leqslant q$ (resp., $|h| > q$). The following lemma and its corollary show that if the target conjunction is long then every long hypothesis has good performance, as both the target and the hypothesis are false on most instances.

Lemma 3.3.1 (Performance Lower Bound). *If* $|h| \geqslant q$ *and* $|c| \geqslant q + 1$ *then* $\mathrm{Perf}_{\mathcal{U}_n}(h, c) > 1 - 3 \cdot 2^{-q}$.

Proof. Apply (3.7) with $m + r = q$, $m + u \geqslant q + 1$. $\mathrm{Perf}_{\mathcal{U}_n}(h, c) > 1 - 2 \cdot 2^{-(m+u)} - 2 \cdot 2^{-(m+r)} \geqslant 1 - 2 \cdot 2^{-q-1} - 2 \cdot 2^{-q} = 1 - 3 \cdot 2^{-q}$ □

Corollary 3.3.2. *Let* $q \geqslant \lg(3/\varepsilon)$. *If* $|h| \geqslant q$, $|c| \geqslant q + 1$ *then* $\mathrm{Perf}_{\mathcal{U}_n}(h, c) > 1 - \varepsilon$.

Proof. By Lemma 3.3.1 $\mathrm{Perf}_{\mathcal{U}_n}(h, c) > 1 - 3 \cdot 2^{-q} \geqslant 1 - 3 \cdot 2^{\lg(\varepsilon/3)} = 1 - \varepsilon$. □

3.3.1 Properties of the Local Search Procedure

Local search, when switching to h' from h, is guided by the quantity

$$\Delta = \mathrm{Perf}_{\mathcal{U}_n}\left(h', c\right) - \mathrm{Perf}_{\mathcal{U}_n}(h, c). \tag{3.8}$$

We analyze Δ using (3.7). The analysis is summarized in Figure 3.1, where the node *good* represents good variables and the node *bad* represents bad variables. Note that Δ depends only on the type of mutation performed and on the values of the parameters m, u and r; in fact, as the analysis shows, it depends on the size of the hypothesis $|h| = m + r$ and on the number u of undiscovered variables.

Comparing $h' \in N^+$ with h. We introduce a variable z in the hypothesis h. If z is good, $\Delta = 2^{-|h|} > 0$. If z is bad, $\Delta = 2^{-|h|}(1 - 2^{1-u})$.

Comparing $h' \in N^-$ with h. We remove a variable z from the hypothesis h. If z is good, $\Delta = -2^{1-|h|} < 0$. If z is bad, $\Delta = -2^{1-|h|}(1 - 2^{1-u})$.

Comparing $h' \in N^{+-}$ with h. Replacing a good with a bad variable gives $\Delta = -2^{1-|h|-u}$. Replacing a good with a good, or a bad with a bad variable gives $\Delta = 0$. Replacing a bad with a good variable gives $\Delta = 2^{2-|h|-u}$.

Correlation produces a perhaps unexpected phenomenon already in the case of the uniform distribution: adding a bad variable can result in Δ being positive, 0 or negative, depending on the number of undiscovered variables.

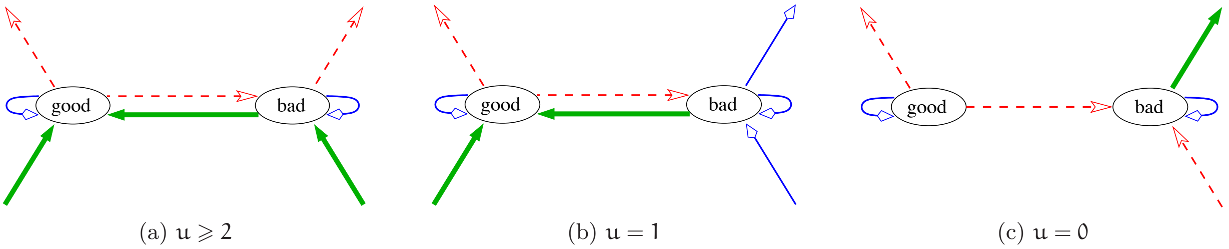


Figure 3.1: Arrows pointing towards the nodes indicate additions of variables and arrows pointing away from the nodes indicate removals of variables. Note that this is consistent with arrows indicating the swapping of variables. Thick solid lines indicate $\Delta > 0$, simple lines indicate $\Delta = 0$, and dashed lines indicate $\Delta < 0$. Usually Figure 3.1a applies. When only one good variable is missing we have the case shown in Figure 3.1b. Once all good variables are discovered, Figure 3.1c applies; hence two arrows disappear. Note that an arrow with $\Delta > 0$ may correspond to a beneficial *or* neutral mutation, depending on the value of the tolerance t.

Now we turn to characterizing the best bounded size approximations of concepts, implicit in the analysis of the swapping algorithm. The existence of such characterizations seems to be related to efficient evolvability and so it may be of interest to formulate it explicitly. Such a characterization does *not* hold for product distributions in general, as noted in the next section. However, as shown in Section 3.5, the analogous characterization *does* hold for every product distribution if the fitness function is changed from correlation to covariance.

Theorem 3.3.3 (Structure of Best Approximations). *The best q-approximation of a target c is c if $|c| \leqslant q$, or any hypothesis formed by q good variables if $|c| > q$.*

Proof. The claim follows directly from the definitions if $|c| \leqslant q$. Let $|c| > q$. Let h be a hypothesis consisting of q good variables. Then both deleting a variable or swapping a good variable for a bad one decrease performance. Thus h cannot be improved among hypotheses of size at most q. If h has fewer than q variables then it can be improved by adding a good variable. If h has q variables but it contains a bad variable then its performance can be improved by swapping a bad variable for a good one. Hence every hypothesis other than the ones described in the theorem can be improved among hypotheses of size at most q. □

3.3.2 Evolving Monotone Conjunctions under the Uniform Distribution

The core of the algorithm for evolving monotone conjunctions outlined in Section 3.1 is composed by the `Mutator` function, presented in Algorithm 1. The role of `Mutator` is, given a current hypothesis h, to produce a new hypothesis h' which has better performance than h if Bene is nonempty, or else a hypothesis h' with about the same performance as h, in which case h' arises from h by a neutral mutation. Hence, during the evolution, we have g calls to `Mutator` throughout a sequence of g generations. We pass some slightly different parameters in the `Mutator` from those defined in [138], to avoid ambiguity. Hence, `Mutator` receives as input q, the maximum allowed size for the approximation, $s_{M,1}$, the sample size used for all the empirical estimates of the performance of each conjunction of size up to q, $s_{M,2}$ the sample size used for conjunctions of length greater than q, and the current hypothesis h. We view conjunctions as objects that have two extra attributes, their *weight* and the *value* of their performance. `GetPerformance` returns the value of the performance, previously assigned by `SetPerformance`. The performance of the initial hypothesis has been determined by another similar call to the `SetPerformance` function with the appropriate sample size. Weights are assigned via

SetWeight. Hence, SetWeight assigns the same weight to all members of $\{h\} \cup N^- \cup N^+$ so that they add up to $1/2$, and the same weight to all the members of N^{+-} so that they add up to $1/2$. Finally, RandomSelect computes the sum W of weights of the conjunctions in the set that it has as argument, and returns a hypothesis h' from that set with probability $w_{h'}/W$, where $w_{h'}$ is the weight of h'.

Note that the neighborhoods and the tolerances are different for short and long hypotheses, where a hypothesis h is short if $|h| \leqslant q$, and

$$q = \left\lceil \lg \frac{3}{\varepsilon} \right\rceil . \tag{3.9}$$

In order to prove Theorem 3.3.8 below we will need the following lemmas.

Lemma 3.3.4 (Lower Bound on Additions for Short Hypotheses). *For the non-zero differences of* Δ *when we add a variable it holds* $|\Delta| \geqslant 2^{-1-m-r}$, *where* $m+r \leqslant q-1$.

Proof. We can add a variable only if $|h| = m + r \leqslant q - 1$. If the variable is good, then $\Delta = 2^{-|h|} = 2^{-m-r} \geqslant 2^{1-q}$. If the variable is bad, then $|\Delta| = |2^{-|h|}(1-2^{1-u})| \geqslant 2^{-m-r} \cdot 2^{-1} = 2^{-1-m-r} \geqslant 2^{-q}$. □

Lemma 3.3.5 (Lower Bound on Removals for Short Hypotheses). *Removing a variable from a hypothesis that has identified all the good variables changes the performance by at least* $|\Delta| = 2^{1-m-r}$, *where* $m + r \leqslant q$.

Proof. We are interested in the case where $u = 0$. Regardless of what the variable is, then $|\Delta| = 2^{1-|h|} = 2^{1-m-r} \geqslant 2^{1-q}$. □

Lemma 3.3.6 (Lower Bound on Swaps for Short Hypotheses and Short Targets). *Let h be a hypothesis that is short. For the beneficial mutations when* $u \geqslant 1$ *as well as all the non-zero differences* Δ *when* $u = 0$ *it holds that* $|\Delta| \geqslant 2^{1-m-r-u}$.

Proof. Swapping a good with a good variable or a bad with a bad we have $\Delta = 0$. Swapping a good with a bad variable gives $|\Delta| = 2^{1-|h|-u} \geqslant 2^{1-q-(q-1)} \geqslant 2^{2-2q}$. Swapping a bad with a good variable gives $\Delta = 2^{2-|h|-u} \geqslant 2^{2-q-q} = 2^{2-2q}$. □

Lemma 3.3.7 (Upper Bound on Δ for Long Hypotheses). *Any mutation of a long hypothesis does not change the performance by more than* 2^{-q}.

Proof. Note, that it holds $|h| \geqslant q+1$. When we remove a good variable, $|\Delta| = 2^{1-|h|} \leqslant 2^{1-(q+1)} = 2^{-q}$. When we remove a bad variable, $|\Delta| = 2^{1-|h|}(1 - 2^{1-u}) \leqslant 2^{1-(q+1)} \cdot 1 = 2^{-q}$. □

Theorem 3.3.8. *For every target conjunction c and every initial hypothesis* h_0 *it holds that after* $\mathcal{O}\left(q + |h_0| \ln \frac{1}{\delta}\right)$ *iterations, each iteration evaluating the performance of* $\mathcal{O}(nq)$ *hypotheses, and each performance being evaluated using sample size* $\mathcal{O}\left(\left(\frac{1}{\varepsilon}\right)^4 \left(\ln n + \ln \frac{1}{\delta} + \ln \frac{1}{\varepsilon}\right)\right)$ *per iteration, equation (3.2) is satisfied.*

Proof. The analysis depends on the size of the target and the initial hypothesis.

Short Initial Hypothesis and Short Target. Note first that for any hypothesis h and for any target c such that $|h|, |c| \leqslant q$, by Lemmas 3.3.4, 3.3.5, and 3.3.6 $|\Delta| \geqslant 2^{2-2q}$. Tolerance for short hypotheses is $t = \frac{1}{2} 2^{2-2q} = 2^{1-2q}$. Hence as long as the estimate of the performance is within t of its exact value, beneficial mutations are identified as beneficial. Therefore it is sufficient to analyze the signs of Δ along the arrows in Figure 3.1. Note that deleting a good variable is always deleterious, and so u is non-increasing.

If there are at least two undiscovered variables (i.e., $u \geqslant 2$, corresponding to Figure 3.1a), then beneficial mutations can only *add* or *swap* variables. Each swap increases the number of good variables, and so after $|c| - 1$ many swaps there is at most one undiscovered variable. Hence, as long as $u \geqslant 2$, there can be at most $q - |h_0|$ additions and at most $|c| - 1$ swaps.

If there is one undiscovered variable (i.e., $u = 1$, corresponding to Figure 3.1b), then, in 1 step, the first beneficial mutation brings this variable into the hypothesis, and all variables become discovered.

If all variables are discovered (i.e., $u = 0$, corresponding to Figure 3.1c) then beneficial mutations are those which delete bad variables from h_0. After we get to the target, there are no beneficial mutations, and the only neutral mutation is the target itself, hence there is no change. Thus the number of steps until getting to the target is at most $q - |c|$.

Summing up the above, the total number of steps is at most $2q - |h_0| \leqslant 2q$.

Short Initial Hypothesis and Long Target. As long as $|h| < q$, we have $u \geqslant 2$, corresponding to Figure 3.1a. Therefore adding any variable is beneficial. Note that replacing a bad variable by a good one may or may not be so, depending on the size of c. The same analysis as above implies that after at most $2q$ beneficial mutations we reach a hypothesis of size q.

If $|c| \geqslant q + 2$ then $u \geqslant 2$ continues to hold, and so all beneficial or neutral mutations will keep hypothesis size at q. However, by Corollary 3.3.2, all those hypotheses have performance at least $1 - \varepsilon$.

If $|c| = q + 1$ then after reaching level q there is one undiscovered variable, corresponding to Figure 3.1b. Swaps of bad variables for good ones are beneficial. Combining these mutations with the ones needed to reach level q, we can bound the *total* number of steps until reaching a hypothesis of q good variables by $2q$ (using the same argument as above). After that, there are only neutral mutations swapping a good variable with another good one, and again all those hypotheses have performance at least $1 - \varepsilon$.

As a summary, if we start from a short hypothesis and all the empirical tests perform as expected, then, we are *always* at a good hypothesis after $2q$ iterations. This will not be the case when we start from a long hypothesis.

Complexity Analysis for Short Hypotheses. From the previous two paragraphs, we have seen that this phase requires at most $2q$ steps regardless whether the target is a short or a long hypothesis. In each of these steps, by Lemma 3.2.1, the algorithm generates a neighborhood no larger than $|N| = (q+1)n + 1 - q^2$. For simplicity we assume $\varepsilon < 3/2$ (i.e. $q = \lceil \lg \frac{3}{\varepsilon} \rceil \geqslant 2$), and hence, $|N| \leqslant 2qn$. This implies that we have to estimate the performance of no more than $(2q) \cdot (2qn) = 4q^2n$ hypotheses, each one of them within accuracy $\epsilon = t = 2^{-2q} \leqslant \varepsilon^2/9$. By the Hoeffding Bound (Proposition 2.2.10) when setting $\alpha = -1, \beta = 1, \epsilon = \varepsilon^2/9$, the performance of each hypothesis is not estimated within $\epsilon = \varepsilon^2/9$ of its true value with probability $e^{-R\varepsilon^4/18}$. By the Union Bound (Proposition 2.2.5) the performance of each hypothesis is computed within $\epsilon = \varepsilon^2/9$ of its true value with probability at least $1 - \sum_{\text{all hypotheses}} e^{-R\varepsilon^4/162} \geqslant 1 - 4q^2ne^{-R\varepsilon^4/162}$. We require now this probability to be at least $1 - \delta/2$ and hence it is enough if each empirical estimate is computed with at least $R \geqslant \left\lceil \frac{162}{\varepsilon^4} \ln\left(\frac{8q^2n}{\delta}\right) \right\rceil$ samples; i.e. we require $\mathcal{O}\left(\frac{1}{\varepsilon^4} \cdot \left(\ln n + \ln \frac{1}{\delta} + \ln \lg^2 \frac{1}{\varepsilon}\right)\right)$ samples for the approximation of the performance of each hypothesis. Hence, the total number of samples for this phase is $\mathcal{O}\left(n \cdot \left(\frac{1}{\varepsilon}\right)^4 \cdot \lg^2 \frac{1}{\varepsilon} \cdot \left(\ln n + \ln \frac{1}{\delta} + \ln \lg^2 \frac{1}{\varepsilon}\right)\right)$.

Long Initial Hypothesis. For long hypotheses the neighborhood consists of hypotheses obtained by deleting a variable, and the hypothesis itself. We set the tolerance in such a way that every hypothesis in the neighborhood is neutral. This guarantees that with high probability in $\mathcal{O}\left(|h_0| \ln \frac{1}{\delta}\right)$ iterations we arrive at a hypothesis of size at most q, and from then on we can apply the analysis of the previous two cases. The model assumes that staying at a hypothesis is always a neutral mutation, hence it is possible to end up in a hypothesis of size bigger than q.

Complexity Analysis for Evolving a Short Hypothesis from a Long Initial Hypothesis. We have to take care of two different phenomena in this phase of the algorithm. One is to guarantee that with high probability after a certain number of generations the algorithm will evolve to a hypothesis with size at most q, and second, that with high probability, every empirical estimate for the performance of each hypothesis in the neighborhood of each generation is computed within accuracy $\epsilon = 2^{-q} = \varepsilon/3$.

Regarding the first phenomenon, and assuming that all the hypotheses in each neighborhood are classified as neutral, then, for a hypothesis h, such that $|h| = k$, there are $k+1$ hypotheses in the neighborhood, and k of them lead to a shorter hypothesis. Hence, with probability $k/(k+1)$ we are led to a hypothesis of smaller size in each step. We need $|h_0| - q \leqslant |h_0|$ such *successes*, each of which happens with probability $\frac{|h_i|}{|h_i|+1} \geqslant \frac{q+1}{q+2} \geqslant \frac{3}{4}$, since $q \geqslant 2$. We apply Lemma 2.2.15 with $p = \frac{3}{4}, \kappa = |h_0|$, and $\delta_C = \frac{\delta}{4}$. We get $t = \left\lceil \frac{8}{3} \cdot \left(|h_0| + \ln\left(\frac{4}{\delta}\right)\right)\right\rceil$; i.e. $\mathcal{O}\left(|h_0| + \ln\left(\frac{1}{\delta}\right)\right)$ generations are enough.

Regarding the second phenomenon, we have a similar argument as earlier. We use Proposition 2.2.10 with $\alpha = -1, \beta = 1, \epsilon = \varepsilon/3$. The performance of any hypothesis is not computed within $\epsilon = 2^{-q} = \varepsilon/3$ of its true value with probability $e^{-R\varepsilon^2/18}$. The entire process lasts for $t = \left\lceil \frac{8}{3} \cdot \left(|h_0| + \ln\left(\frac{4}{\delta}\right)\right)\right\rceil$ steps, and in each step the neighborhood has size $|h_i| + 1 \leqslant n + 1$. By the Union Bound (Proposition 2.2.5) the probability that any empirical estimate is not within $\epsilon = \varepsilon/3$ of its true value is at most $(n+1) \cdot t \cdot e^{-R\varepsilon^2/18}$. We now require to bound this quantity from above by $\delta/4$ and hence we need $R \geqslant \left\lceil \frac{18}{\varepsilon^2} \cdot \ln\left(\frac{4(n+1)t}{\delta}\right)\right\rceil$ samples per empirical estimate computation; that is, we need $\mathcal{O}\left(\left(\frac{1}{\varepsilon}\right)^2 \left(\ln n + \ln|h_0| + \ln\frac{1}{\delta}\right)\right)$ samples for the approximation of the performance of each hypothesis. As a consequence, the total number of samples for this phase is $\mathcal{O}\left(n \cdot \left(|h_0| + \ln\frac{1}{\delta}\right) \cdot \left(\frac{1}{\varepsilon}\right)^2 \cdot \left(\ln n + \ln|h_0| + \ln\frac{1}{\delta}\right)\right)$.
□

3.4 Monotone Conjunctions under Product Distributions using Correlation

A *product distribution* over $\{0,1\}^n$ is specified by the probabilities $p = (p_1, \ldots, p_n)$, where p_i is the probability of setting the variable x_i to 1. The probability of a truth assignment $(a_1, \ldots, a_n) \in \{0,1\}^n$ is $\prod_{i=1}^n p_i^{a_i} \cdot (1-p_i)^{1-a_i}$. For the uniform distribution $\mathcal{U}_n$ the probabilities are $p_1 = \ldots = p_n = 1/2$. We write $\mathcal{P}_n$ to denote a fixed product distribution, omitting p for simplicity.

Let us consider a target c and a hypothesis h as in (3.6). Let $\textsc{index}(z)$ be a function that returns the set of indices of the participating variables in a hypothesis z. We define the sets $\mathfrak{M} = \textsc{index}(h) \cap \textsc{index}(c)$, $\mathfrak{R} = \textsc{index}(h) \setminus \mathfrak{M}$, and $\mathfrak{U} = \textsc{index}(c) \setminus \mathfrak{M}$. We can now define

$$M = \prod_{i \in \mathfrak{M}} p_i, \qquad R = \prod_{\ell \in \mathfrak{R}} p_\ell, \qquad \text{and} \qquad U = \prod_{k \in \mathfrak{U}} p_k.$$

Finally, set $|\mathfrak{M}| = m, |\mathfrak{R}| = r$, and $|\mathfrak{U}| = u$. Then (3.7) generalizes to

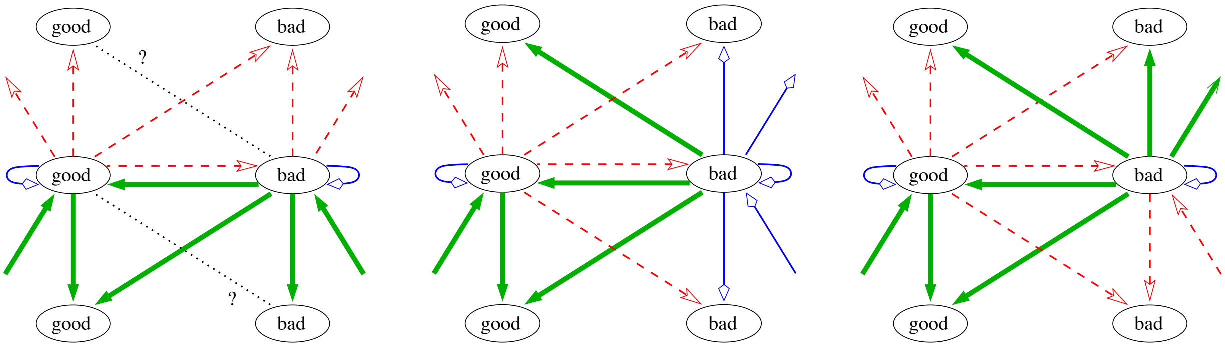
$$\mathrm{Perf}_{\mathcal{P}_n}(h, c) = 1 - 2M(R + U - 2RU). \tag{3.10}$$

We impose some conditions on the p_i's in the product distribution.

Definition 3.4.1 (Nondegenerate Product Distribution)**.** A product distribution $\mathcal{P}_n$ given by $p = (p_1, \ldots, p_n)$ is μ-nondegenerate if

- $\min\{p_z, 1-p_z\} \geqslant \mu$ for every variable z
- the difference of any two members of the multiset $\{p_1, 1-p_1, \ldots, p_n, 1-p_n\}$ is zero, or has absolute value at least μ.

The following lemma and its corollary are analogous to Lemma 3.3.1 and Corollary 3.3.2.



(a) $U < 1/2$ (b) $U = 1/2$ (c) $U > 1/2$

Figure 3.2: The style and the directions of arrows have the same interpretation as in Figure 3.1. The middle layer represents variables that have the same probability of being satisfied under the distribution; i.e. $p_{\text{good}} = p_{\text{bad}}$. A node x that is one level above another one y indicates higher probability of satisfying the variable x; i.e. $p_x > p_y$. Here we distinguish the three basic cases for U; for two arrows in the first case we have a ? to indicate that Δ can not be determined by simply distinguishing cases for U.

Lemma 3.4.2 (Performance Lower Bound). *Let a hypothesis h such that* $|h| \geqslant q-1$ *and a target c such that* $|c| \geqslant q+1$*. Then,* $Perf_{\mathcal{P}_n}(h,c) > 1 - 6.2 \cdot e^{-\mu q}$.

Proof. The setup of the lemma implies $m + r \geqslant q - 1$, and $m + u \geqslant q + 1$. Using (3.10) we have:

$$\begin{aligned}
\mathrm{Perf}_{\mathcal{P}_n}(h,c) &> 1 - 2MR - 2MU && \text{(by (3.10))}\\
&\geqslant 1 - 2e^{-\mu(m+r)} - 2e^{-\mu(m+u)} && \text{(Definition 3.4.1)}\\
&\geqslant 1 - 2e^{-\mu q}e^{\mu} - 2e^{-\mu q}e^{-\mu} && (m+r \geqslant q-1, m+u \geqslant q+1)\\
&= 1 - 4e^{-\mu q}\cosh(\mu) && (e^{\mu} + e^{-\mu} = 2\cosh(\mu))
\end{aligned}$$

Since $1 \leqslant \cosh(\mu) < 1.55\ \forall \mu \in [0,1]$, we have $\mathrm{Perf}_{\mathcal{P}_n}(h,c) > 1 - 6.2e^{-\mu q}$. □

Corollary 3.4.3. *Let* $q \geqslant \frac{1}{\mu}\ln\left(\frac{6.2}{\varepsilon}\right)$, $|h| \geqslant q-1$, $|c| \geqslant q+1 \Rightarrow Perf_{\mathcal{U}_n}(h,c) > 1 - \varepsilon$.

Proof. By Lemma 3.4.2 $\mathrm{Perf}_{\mathcal{P}_n}(h,c) > 1 - 6.2 \cdot e^{-\mu q} \geqslant 1 - 6.2 \cdot e^{-\mu\mu^{-1}\ln(6.2/\varepsilon)} = 1 - 6.2 \cdot e^{\ln(\varepsilon/6.2)} = 1 - \varepsilon$. □

3.4.1 Properties of the Local Search Procedure

We want to generalize the results of Section 3.3.1 by looking at the quantity

$$\Delta = \mathrm{Perf}_{\mathcal{P}_n}(h', c) - \mathrm{Perf}_{\mathcal{P}_n}(h, c) \tag{3.11}$$

which corresponds to (3.8). We use (3.10) for the different types of mutations.

The signs of Δ depend on the ordering of the probabilities p_i. A variable x_i is *smaller* (resp., *larger*) than a variable x_j if $p_i < p_j$ (resp., $x_i > x_j$). If $p_i = p_j$ then x_i and x_j are *equivalent*. Analyzing Δ, we draw the different cases in Figure 3.2. However, when $U < 1/2$, two arrows can not be determined. These cases refer to mutations where we replace a bad variable with a bigger good one, or a good variable with a smaller bad one. Both mutations depend on the distribution; the latter has $\Delta = -2MR(p_{\mathbf{in}}/p_{\mathbf{out}} - 1 + 2U(1 - p_{\mathbf{in}}))$, where **out** is a good variable and **in** is the bad smaller variable. One application of this equation is that the Structure Theorem 3.3.3 does not hold under product distributions. The other application is the construction of local optima.

The first idea on constructing local optima is to require the difference Δ_2 that is achieved when we insert a *bad* variable to be more than the difference Δ_1 that is achieved when we insert a *good* variable. For simplicity, and reasons that will be apparent later on, assume all the good literals have probability g of being satisfied and all the bad literals have probability b of being satisfied. Implementing the above idea we require $2MR(1-b)(1-2U) > 2MR(1-g)$. Simplifying, for u undiscovered literals, this reduces to:

$$b < g \cdot \frac{1-2g^{u-1}}{1-2g^{u}} \tag{3.12}$$

By (3.12) we have that for $u \geqslant 2$ there are values of b and g such that the inequality is satisfied. Also note that as u increases, then the right hand side of (3.12) also increases. Hence, from now on we will focus on $u = 2$. We require

$$b < g \cdot \frac{1-2g}{1-2g^{2}}. \tag{3.13}$$

For $x_0 = 1 - \sqrt{2}/2 \approx 0.2929$, the function $f(x) = x(1-2x)/(1-2x^2)$ achieves its maximum. In order to keep the numbers simple we can set $g = 1/3$ and $b = 1/10$. This implies that if there are at least two undiscovered good variables, adding a bad variable is more beneficial than adding a good one. Hence, for short targets, whenever we have a hypothesis h which misses at least two good variables, the algorithm will be more inclined, than not, to introduce a bad variable instead of a good one, and therefore will go towards local optima as long as there are at least $q+1$ bad variables in our domain.

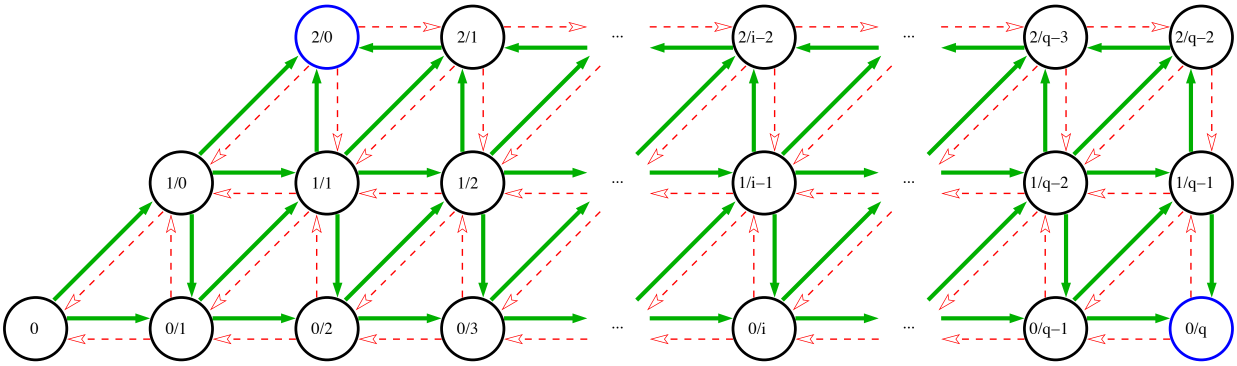


Figure 3.3: The lattice where the search is performed. In every state there is another arrow leading to the same state, representing neutral mutations with the same amount and quality (i.e. good and bad) of variables. This arrow is not drawn for clarity.

Performance with 0 Good and i Bad Variables. Any hypothesis h with 0 good and i bad variables has performance

$$\begin{aligned} \mathrm{Perf}_{\mathcal{P}_n}(h,c) &= 1 - 2(10^{-i} + 3^{-2} - 2\cdot 10^{-i}3^{-2}) \\ &= \frac{7}{9} - \frac{14}{9}10^{-i} \end{aligned} \tag{3.14}$$

Hence, as $i \uparrow \Longrightarrow \mathrm{Perf}_{\mathcal{P}_n}(h,c) \uparrow$.

Performance with 1 Good and i Bad Variables. Any hypothesis h with 1 good and i bad variables has performance

$$\begin{aligned} \mathrm{Perf}_{\mathcal{P}_n}(h,c) &= 1 - 2\cdot 3^{-1}(10^{-i} + 3^{-1} - 2\cdot 10^{-i}3^{-1}) \\ &= \frac{7}{9} - \frac{2}{9}10^{-i} \end{aligned} \tag{3.15}$$

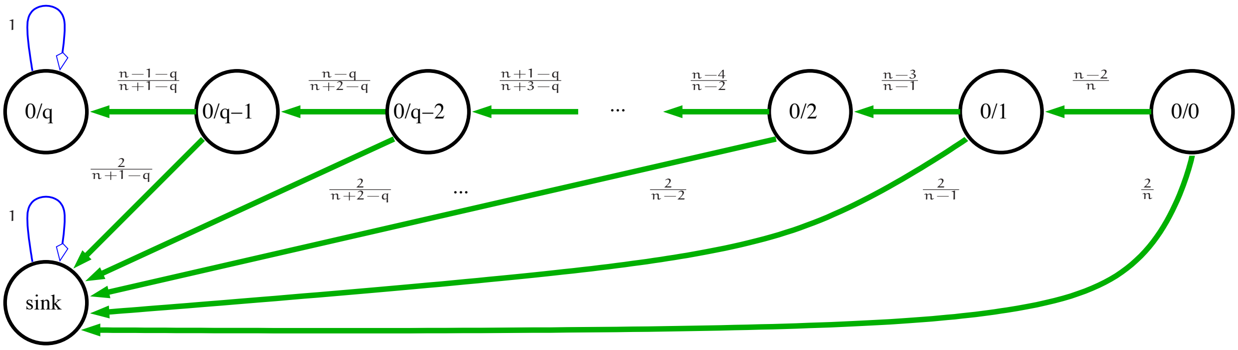


Figure 3.4: When tolerance is small and everything is determined.

Hence, as $i \uparrow \Longrightarrow \mathrm{Perf}_{\mathcal{P}_n}(h, c) \uparrow$.

Performance with 2 Good and i Bad Variables. Any hypothesis h with 2 good and i bad variables has performance

$$\begin{aligned} \mathrm{Perf}_{\mathcal{P}_n}(h, c) &= 1 - 2 \cdot 3^{-2}(10^{-i} + 1 - 2 \cdot 10^{-i}) \\ &= \frac{7}{9} + \frac{2}{9} 10^{-i} \end{aligned} \tag{3.16}$$

Hence, as $i \uparrow \Longrightarrow \mathrm{Perf}_{\mathcal{P}_n}(h, c) \downarrow$.

The transition probability matrix (in canonical form) is

$$P = \left[\begin{array}{c|c} I & \mathbb{O} \\ \hline R & Q \end{array}\right] = \left[\begin{array}{cc|ccccc} 1 & 0 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & 1 & 0 & \cdots & \cdots & \cdots & 0 \\ \hline \frac{n-1-q}{n+1-q} & \frac{2}{n+1-q} & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \frac{2}{n+2-q} & \frac{n-q}{n+2-q} & 0 & & & \vdots \\ 0 & \frac{2}{n+3-q} & 0 & \frac{n+1-q}{n+3-q} & 0 & & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \frac{2}{n} & 0 & \cdots & 0 & \frac{n-2}{n} & 0 \end{array}\right]. \tag{3.17}$$

The fundamental matrix (see [77] for the definition) of the chain is

$$N = (I - Q)^{-1} = \begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ \frac{n-q}{n+2-q} & 1 & 0 & \cdots & \cdots & 0 \\ \prod_{j=2}^{3} \frac{n-2-(q-j)}{n-(q-j)} & \frac{n+1-q}{n+3-q} & 1 & 0 & \cdots & 0 \\ \prod_{j=2}^{4} \frac{n-2-(q-j)}{n-(q-j)} & \prod_{j=3}^{4} \frac{n-2-(q-j)}{n-(q-j)} & \frac{n+2-q}{n+4-q} & 1 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\ \prod_{j=2}^{q} \frac{n-2-(q-j)}{n-(q-j)} & \prod_{j=3}^{q} \frac{n-2-(q-j)}{n-(q-j)} & \prod_{j=4}^{q} \frac{n-2-(q-j)}{n-(q-j)} & \cdots & \frac{n-3}{n-1} & 1 \end{bmatrix}. \tag{3.18}$$

The absorption probabilities are given by the matrix

$$B = N \cdot R = \left[\begin{array}{c|c} \frac{n-1-q}{n+1-q} & \frac{2}{n+1-q} \\ \prod_{j=1}^{2} \frac{n-2-(q-j)}{n-(q-j)} & 1 - \prod_{j=1}^{2} \frac{n-2-(q-j)}{n-(q-j)} \\ \prod_{j=1}^{3} \frac{n-2-(q-j)}{n-(q-j)} & 1 - \prod_{j=1}^{3} \frac{n-2-(q-j)}{n-(q-j)} \\ \vdots & \vdots \\ \prod_{j=1}^{q} \frac{n-2-(q-j)}{n-(q-j)} & 1 - \prod_{j=1}^{q} \frac{n-2-(q-j)}{n-(q-j)} \end{array}\right] \tag{3.19}$$

Note that the closed form products in the first column are formed by multiplying positive numbers which are strictly less than 1; hence, the products obtain smaller values as more and more terms are included. Therefore, starting from any hypothesis that does not contain any good variables, the swapping algorithm will reach a local optimum in at most q steps with probability $\mathbf{Pr}\left(\textsc{Local Optimum}\right)$ at least

$$\prod_{j=1}^{q}\frac{n-2-(q-j)}{n-(q-j)}=\prod_{r=0}^{q-1}\left(1-\frac{2}{n-r}\right)=\frac{(n-q)(n-q-1)}{n(n-1)}\,. \tag{3.20}$$

Moreover, $\lim_{n\to\infty}\mathbf{Pr}\left(\textsc{Local Optimum}\right)=1$. The example below gives a summary.

Example 1. Let $\mathcal{P}_n$ be a distribution such that $p_1=p_2=1/3$, and the rest of the $n-2$ variables are satisfied with probability $1/10$. Set the target $c=x_1\wedge x_2$. A hypothesis h formed by q bad variables has performance $\mathrm{Perf}_{\mathcal{P}_n}(\mathrm{h},\mathrm{c})=1-2\mathbf{Pr}\left(\text{error region}\right)<1-2/9=7/9$. Note that, for the nonzero values of Δ, it holds $|\Delta|\geqslant 2\mu^{q+2}$. Hence, by setting the tolerance $t=\mu^{q+2}$, and the accuracy on the empirical tests on conjunctions of size at most q, equal to $\epsilon_M=t=\mu^{q+2}$, all the arrows in the diagrams can be determined precisely.

Let $\rightsquigarrow$ denote a mutation. Starting from $\mathrm{h}_0=\emptyset$, there are sequences of beneficial mutations in which the algorithm inserts a bad variable in each step, e.g. $h_0=\emptyset\rightsquigarrow h_1=x_3\rightsquigarrow\ldots\rightsquigarrow h_q=\bigwedge_{\ell=3}^{q+2}x_\ell$. This is a local optimum, since swapping a bad variable with a good one yields $\Delta<0$. Note that $\mu=1/10, q=\lceil 10\ln(62)\rceil=42$, and for $\varepsilon=1/10$ the algorithm is stuck in a hypothesis with $\mathrm{Perf}_{\mathcal{P}_n}(\mathrm{h}_q,\mathrm{c})<1-\varepsilon$.

Under the setup of the example above, the algorithm will insert q bad variables in the first q steps, with probability $\Gamma=\prod_{r=0}^{q-1}\left(1-\frac{2}{n-r}\right)=\frac{(n-q)(n-q-1)}{n(n-1)}$. Requiring $n\geqslant\left\lceil\frac{2q}{\delta}\right\rceil$ we have $\Gamma\geqslant 1-\delta$. Hence, starting from the empty hypothesis, the algorithm will fail for *any* $\varepsilon<2/10$, with probability 0.9, if we set $n\geqslant 840$.

3.4.2 Special Cases

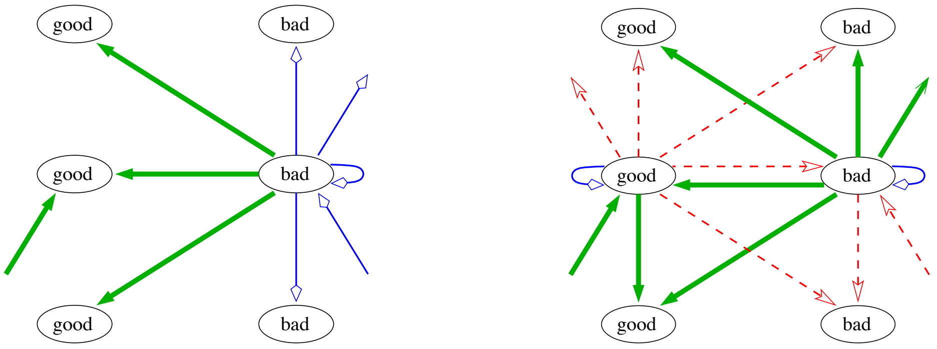
Although for arbitrary targets and arbitrary product distributions we can not guarantee that the algorithm will produce a hypothesis h such that (3.2) is satisfied, we can however, pinpoint some cases where the algorithm will succeed with the correct setup. These cases are targets of size at most 1 or greater than $q+\frac{1}{\mu}\ln 2$, and *heavy* targets; i.e. targets that are satisfied with probability at least $1/2$ which are presented below.

Heavy Targets (Special Case of Short Targets)

Heavy targets are targets such that $\prod p_j\geqslant\frac{1}{2}$. A first consequence of this definition is that the target is composed by a few variables, since the weight of any target with k variables is upper bounded by the quantity $e^{-\mu k}$. Hence, requiring $e^{-\mu k}\geqslant 1/2\Rightarrow k\leqslant\frac{1}{\mu}\ln 2$. The second consequence is that for *any* hypothesis the weight of the subcube of the undiscovered variables has weight at least $1/2$. Hence, Figures 3.5a and 3.5b capture the entire evolution process. These figures that dictate the local search are identical to the case where we use the covariance as a fitness function; see Figures 3.9a and 3.9b. Hence, in order to show that the bounds presented in Section 3.5 also hold here as upper bounds, we have to show that the beneficial set for short hypotheses is nonempty with a polynomial tolerance in the case of the mutations that introduce a good variable; either by adding a good variable or swapping a bad variable to a good one. We have

Adding a Good Variable w. In this case the difference in performance is $\Delta=2MR(1-p_w)>2\cdot\frac{1}{2}\cdot R\cdot\mu=R\mu\geqslant\mu^{q-1}\mu=\mu^q$.

Swapping variables.



(a) $U = 1/2$

(b) $U > 1/2$

Figure 3.5: The style and the directions of arrows have the same interpretation as earlier. Same is true about the probability of satisfying good or bad variables depending on their height.

Bad $y \to$ Good w: In this case the difference in performance is $\Delta = 4MRU(1/p_y - 1) + 2MR(1 - p_w/p_y)$, and since $U \geqslant 1/2$, we have $\Delta \geqslant 2MR \cdot \frac{1-p_w}{p_y} \geqslant 2\mu^q\mu = 2\mu^{q+1}$.

Good $w \to$ Bad y: This is the inverse process from above. Here we can use the following argument. Consider the sequence of mutations m_1, m_2 such that $m_1 := y \to w$ and $m_2 := w \to y$; i.e. $h \xrightarrow{m_1} h_1 \xrightarrow{m_2} h_2 = h$. Then, clearly after the application of these two mutations there has been no change in the performance of the hypothesis. However, it holds

$$\begin{cases} \mathrm{Perf}_{\mathcal{P}_n}(h_1, c) &= \mathrm{Perf}_{\mathcal{P}_n}(h, c) + \Delta_1 \\ \mathrm{Perf}_{\mathcal{P}_n}(h_2, c) &= \mathrm{Perf}_{\mathcal{P}_n}(h_1, c) + \Delta_2 \end{cases}$$

Adding the above two, we get $\mathrm{Perf}_{\mathcal{P}_n}(h_2, c) = \mathrm{Perf}_{\mathcal{P}_n}(h, c) + \Delta_1 + \Delta_2$. However, since $\mathrm{Perf}_{\mathcal{P}_n}(h_2, c) = \mathrm{Perf}_{\mathcal{P}_n}(h, c)$, we obtain $\Delta_2 = -\Delta_1$. Hence, the lower bound that we have for Δ_1 from the previous case, is a lower bound on the absolute value of Δ_2 for any mutation m_2.

Good out $\to$ Good in: We have $\Delta = 2M \cdot \left(1 - \frac{p_{\mathbf{out}}}{p_{\mathbf{in}}}\right) \cdot (R + U)$. Hence, for the non-zero values of Δ (i.e. $p_{\mathbf{out}} \neq p_{\mathbf{in}}$) and since $R + U \geqslant R \geqslant \mu^q$, we have $|\Delta| \geqslant 2 \cdot \frac{1}{2} \cdot \mu \cdot \mu^q = \mu^{q+1}$.

Bad out $\to$ Bad in: In this case we have $\Delta = 2MR \cdot \left(\frac{p_{\mathbf{in}}}{p_{\mathbf{out}}} - 1\right) \cdot (2U - 1)$. The non-zero values of Δ are obtained when both $p_{\mathbf{in}} \neq p_{\mathbf{out}}$ and $U \neq 1/2$. However, for a hypothesis that does not contain any good variables, the quantity U can be arbitrarily close to $1/2$. Hence, a beneficial such mutation may be characterized as neutral, since U can be super-polynomially away from $1/2$. On the other hand, this is the only case, when a beneficial mutation is not characterized as beneficial. Let s be a positive super-polynomially small quantity close to zero. Then, for targets whose weight can be arbitrarily close to $1/2$, when the hypothesis contains at least one good variable **in**, then it holds $2U' - 1 = 2\frac{U}{p_{\mathbf{in}}} - 1 = 2\frac{\frac{1}{2}+s}{p_{\mathbf{in}}} - 1 \geqslant \frac{1}{p_{\mathbf{in}}} - 1 = \frac{1-p_{\mathbf{in}}}{p_{\mathbf{in}}} \geqslant \frac{\mu}{1-\mu} > \mu$. In other words, we can give a lower bound for this quantity only when at least one good variable appears in the hypothesis. By Lemma 3.5.3 and the above observation we have $|\Delta| > 2\mu^q\mu\mu = 2\mu^{q+2}$.

Concluding, the analysis is similar to that of the case where covariance is used as the fitness function; again in the first step of short hypotheses we bring one good variable in the hypothesis (in case there is none), and from that point on, all the beneficial arrows in the Figures 3.5a and 3.5b are characterized as such. Hence, with a similar analysis, in $\mathcal{O}\left(n^2\right)$ steps in the worst case, the hypothesis will evolve to the target, or to an optimal q-approximation of the target. Here, the term *optimal* is used to indicate that it will satisfy the Structure Theorem 3.5.6. From that point and on, deleting a good variable, or swapping a good variable to a bad one will be characterized as deleterious. Hence, the only neutral mutations, are the ones which swap a good variable to another good variable and at the same time keep the value of the product $\prod_{i \in \mathfrak{M}} p_i$ unchanged (minimum). All these will happen if we set the tolerance for the neighborhood equal to μ^{q+2}.

As a note here, of course, now that we know the lower bounds in the analysis above, we can set the tolerance equal to $\frac{1}{2}\mu^{q+1}$, which will maintain all the good mutations that arise in the beneficial sets, and possibly excluding some of the bad swaps that are characterized as beneficial with the previous setup. This will hopefully save some time in the convergence to the target, however, in terms of worst case analysis we can say nothing more.

Empty Targets

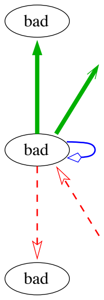


Figure 3.6: The diagram in the case of the empty target.

In this case there are no good variables, and hence it holds $M = U = 1$. Hence, any hypothesis is composed only by redundant variables, and as a consequence the performance of such a function is $\mathrm{Perf}_{\mathcal{P}_n}(h, c) = -1 + 2R$. Again, we look on the differences $\Delta = \mathrm{Perf}_{\mathcal{P}_n}(h', c) - \mathrm{Perf}_{\mathcal{P}_n}(h, c)$, and the results are shown on the diagram on the side. Moreover, we are also interested in lower bounds for the mutations that arise for hypotheses up to size q, as well as an upper bound for the performance fluctuation for hypotheses larger than q. We have:

Adding a Variable z: In this case $\Delta = -2R(1 - p_z)$. Hence, $|\Delta| \geqslant 2\mu^{q-1}\mu = 2\mu^q$.

Removing a Variable z: In this case $\Delta = 2R(1/p_z - 1) > 2\mu^q\mu = 2\mu^{q+1}$.

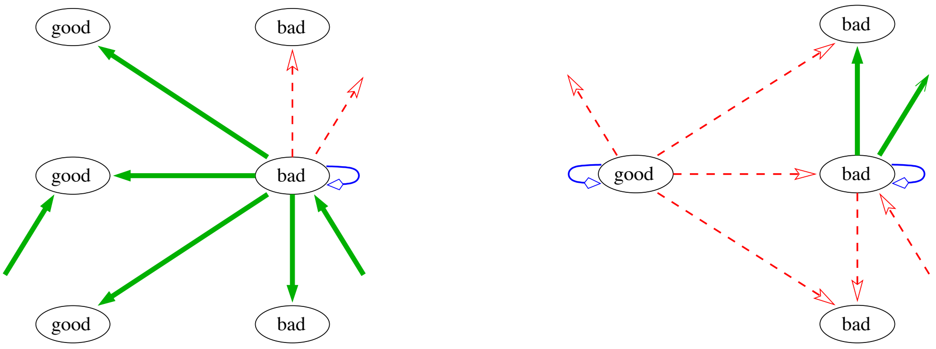
Swapping Variables: A variable z is introduced in place of w. Then, $\Delta = 2R\left(\frac{p_z}{p_w} - 1\right)$. For the non-zero values of Δ (i.e. $p_z \neq p_w$) it holds $|\Delta| > 2\mu^q\mu = 2\mu^{q+1}$.

Short Initial Hypothesis. Hence, if we use tolerance $t = \frac{1}{2} \cdot 2\mu^{q+1} = \mu^{q+1}$, and we approximate the performance of each hypothesis with accuracy $\epsilon = t = \mu^{q+1}$, then, with high probability all the mutations will be characterized correctly. The analysis is similar to that of the proof of covariance, and in $\mathcal{O}\left(n^2\right)$ steps all the bad variables will have been removed and our hypothesis will be the empty conjunction as desired.

Long Initial Hypothesis. The arguments are similar to those in the previous sections and in $\mathcal{O}\left(|h_0| + \ln\frac{1}{\delta}\right)$ we will form a hypothesis of size at most q with probability $1 - \delta/2$. Then, we apply the analysis above.

Targets of Size 1

In this case the target is $c = x$. We assume that $p_x < 1/2$, since otherwise, c is a *heavy* target, and the analysis of Section 3.4.2 can be used to identify the target. Essentially, there are two cases; either the hypothesis contains the good variable x, or it does not.



(a) $U = p_x < 1/2 \ (M = 1)$ (b) $U = 1 \ (M = p_x)$

Figure 3.7: The style and the directions of arrows have the same interpretation as earlier. Same is true about the probability of satisfying good or bad variables depending on their height.

Adding a Variable. If the variable is good (x), then $\Delta = 2R(1 - p_x) \Rightarrow \Delta \geqslant 2 \cdot \mu^{q-1} \cdot \frac{1}{2} = \mu^{q-1}$.

On the other hand, if the added variable is bad (y), then $\Delta = 2R(1 - p_y)(M - 2p_x)$. Hence, if x is undiscovered, $M = 1$, and we can not give a lower bound on Δ, although $\Delta > 0$. On the other hand, if x is already in the hypothesis, then $M = p_x$, and $\Delta = -2R(1 - p_y)p_x < 0$. In this case, $|\Delta| \geqslant 2\mu^{q-1}\mu\mu = 2\mu^{q+1}$.

Deleting a Variable. If the deleted variable is good, then, $\Delta = -2R(1 - p_x) < 0$. Moreover, $|\Delta| \geqslant 2\mu^q\mu = 2\mu^{q+1}$. However, this is a looser lower bound, and we can use the *cyclic* argument, in which case the lower bound becomes μ^{q-1}.

If the deleted variable is bad, then, $\Delta = -2R\left(\frac{1}{p_y} - 1\right)(M - 2p_x)$. If x is contained in the hypothesis, $M = p_x$ (Figure 3.7b), and $\Delta > 2\mu^q\mu\mu = 2\mu^{q+2}$. Again, here a cyclic argument, together with the lower bound on the equivalent insertion above, gives a better bound. On the other hand, if x is not contained in the hypothesis, then $M = 1$ (Figure 3.7a), and $\Delta = -2R\left(\frac{1}{p_y} - 1\right)(1 - 2p_x) < 0$. However, again this time we can not give a lower bound on $|\Delta|$.

Swapping Variables.

Good $x \to$ Bad y: In this case $\Delta = -2R(1 - p_x) - 4Rp_x(1 - p_y) < 0$. Moreover, $|\Delta| > 2R(1 - p_x) \geqslant 2 \cdot \mu^q \cdot \frac{1}{2} = \mu^q$.

Bad $y \to$ Good x: In this case $\Delta = 2\frac{R}{p_y} \cdot (p_x(1 - p_y) + p_y(1 - p_x)) > 0$. However, a good lower bound for such a mutation is accomplished with a cyclic argument and the lower bound in the case above.

Bad $y \to$ Bad z: In this case it holds $\Delta = -2 \cdot R \cdot \left(\frac{p_z}{p_y} - 1\right) \cdot (M - 2p_x)$. If $p_z = p_y$, then $\Delta = 0$ in any case. If $p_z \neq p_y$, then we have to distinguish cases on M. If $M = 1$, then $\Delta > 0$ if $p_z < p_y$, and $\Delta < 0$ if $p_z > p_y$. Note however, that in neither of these cases we can give a lower bound on Δ, since the quantity $(1 - 2p_x)$ can be arbitrarily close to 0. On the other hand, if $M = p_x$, then $\Delta = 2 \cdot R \cdot p_x \cdot \left(\frac{p_z}{p_y} - 1\right)$, in which case $\Delta > 0$ if $p_z > p_y$, and $\Delta < 0$ if $p_z < p_y$. Moreover, in this case we can give a lower bound. In particular, $|\Delta| \geqslant 2\mu^q\mu\mu = 2\mu^{q+2}$.

Complexity Analysis for Short Initial Hypothesis. We set the tolerance (at most) equal to $\frac{1}{2}\mu^q$. Then, in $\mathcal{O}\left(n^2\right)$ generations all possible *bad* $\to$ *bad* beneficial swaps as well as beneficial insertions of bad variables are exhausted, and the single good variable is inserted in the hypothesis. Note that adding the good variable or swapping a bad variable to the single good one is always identified as a beneficial mutation.

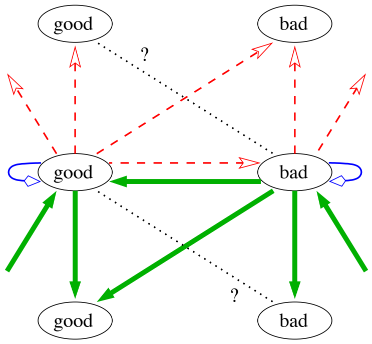


Figure 3.8: The diagram above is the same as in Figure 3.2a.

We are now in Figure 3.7b, and after at most $\mathcal{O}\left(n^2\right)$ generations all possible beneficial mutations are exhausted and we have formed the required hypothesis. Note that for all the deletions of bad variables we have a lower bound by the previous analysis.

Hence, in total we require $\mathcal{O}\left(n^2\right)$ generations for this part.

Complexity Analysis for Long Initial Hypothesis. The argument is similar like in every other case and in $\mathcal{O}\left(|h_0| + \ln\frac{1}{\delta}\right)$ we will form a hypothesis of size at most q with probability $1 - \delta/2$. Then, we apply the analysis above.

Long Targets of Size Greater Than $q + \frac{1}{\mu}\ln 2$

In this case we deal with targets c such that $|c| > q + \frac{1}{\mu}\ln 2$. This implies that any hypothesis of size at most q has $u \geqslant |c| - q > \frac{1}{\mu}\ln 2$ undiscovered good variables. As a consequence, for the weight of the undiscovered cube U for short hypotheses it holds $U \leqslant (1-\mu)^u \leqslant e^{-\mu u} < e^{-\mu\cdot\frac{1}{\mu}\ln 2} = \frac{1}{2}$.

Hence, the diagram that guides the search is shown in Figure 3.8, which is the same as Figure 3.2a.

The idea is that adding a good variable in the hypothesis is always a beneficial mutation, and hence, after a sufficient amount of time, with high probability, the hypothesis can not contain less than $q-1$ variables. As a consequence, such a hypothesis, by Corollary 3.4.3 satisfies the goal of (3.2).

Short Initial Hypothesis We now prove that adding a good variable in a hypothesis of size at most $q-1$ is always beneficial, and in fact we can give a lower bound on the increase in performance in this case.

Adding a Good Variable z. It holds $\Delta = 2MR(1-p_z) \geqslant 2\mu^{|h|}\mu \geqslant 2\mu^{q-1}\mu = 2\mu^q > 0$. This result holds regardless of the case we have shown in Figures 3.8.

Adding a Bad Variable z. It holds $\Delta = 2MR(1-p_z)(1-2U)$. Hence, in the case of Figure 3.8 we have $\Delta > 0$.

Complexity Analysis for Short Initial Hypothesis. The first goal is to form a hypothesis with at least $q-1$ variables. At every step of this part of the evolution, the insertion of good variables is characterized as beneficial with high probability if we set the tolerance equal to $t = \mu^q$. As long as $|h| \leqslant q-2$ there are at least 3 good variables in the beneficial set that arises due to additions of variables. That set has cumulative probability $\frac{1}{2}\cdot\frac{|N^+|}{|N^+|+1} \geqslant \frac{1}{2}\cdot\frac{3}{4} = \frac{3}{8}$. Also note, that removals of variables are not characterized as beneficial. Hence, by Lemma 2.2.15 for $p = \frac{3}{8}$, $\kappa = q-1$, and $\delta_C = \delta/4$ after $t = \left\lceil\frac{16}{3}\left(q - 1 + \ln\left(\frac{4}{\delta}\right)\right)\right\rceil = \mathcal{O}\left(\frac{1}{\mu}\ln\frac{1}{\varepsilon} + \ln\frac{1}{\delta}\right)$ iterations, there have been performed $q-1$ additions of variables with high probability.

Once the hypothesis contains $q-1$ variables, it will keep on oscillating between hypotheses of size $q-1$ and q. The reason is that when a hypothesis is consisted of $q-1$ variables then the beneficial set is non-empty and is formed by additions of variables and possibly swaps, but not with removals of variables, and hence the hypothesis can not shrink. For hypotheses of size q it may be the case that a removal of a bad variable is classified as neutral when the search is guided by the neutral set and hence the size of the hypothesis drops to $q-1$.

Long Initial Hypothesis. Below we examine the complexity starting from a long initial hypothesis.

Complexity Analysis for Long Initial Hypothesis. The argument is similar like in every other case and in $\mathcal{O}\left(|h_0| + \ln\frac{1}{\delta}\right)$ we will form a hypothesis of size at most q with probability $1-\delta/2$. Then, we apply the analysis above, where the hypothesis always has size between $q-1$ and q.

3.5 Covariance as a Fitness Function

The discussion in the previous section shows that there are problems with extending the analysis of the swapping algorithm from the uniform distribution to product distributions. In this section we explore the possibilities of handling product distributions with a different fitness function, covariance, given by (3.5).

Using the same notation as in (3.6), and with $\mathfrak{M}, \mathfrak{R}$, and $\mathfrak{U}$ representing the sets of indices as in the previous section, the first term is given by (3.10). Furthermore,

$$\begin{cases} \mathbf{E}\left[h\right] &= & -1 + 2\cdot\prod_{i\in\mathfrak{M}} p_i \cdot \prod_{\ell\in\mathfrak{R}} p_\ell &=& -1+2MR \\ \mathbf{E}\left[c\right] &= & -1 + 2\cdot\prod_{i\in\mathfrak{M}} p_i \cdot \prod_{k\in\mathfrak{U}} p_k &=& -1+2MU \end{cases} \tag{3.21}$$

Thus from (3.10) and (3.21) we get

$$\mathbf{Cov}\left[h, c\right] = 4MRU\left(1-M\right). \tag{3.22}$$

Note that $\mathbf{Cov}\left[h,c\right] = 4MRU(1-M) \in [0,1]$.

Lemma 3.5.1 (Maximality of Target). $(\forall h \neq c)\left[\mathbf{Cov}\left[h,c\right] < \mathbf{Cov}\left[c,c\right]\right].$

Proof. For $h \neq c$, $\mathbf{Cov}\left[h,c\right] = 4MRU(1-M) < 4M(1-M) = \mathbf{Cov}\left[c,c\right].$ □

Lemma 3.5.2 (Trivial Targets). *Approximating targets c such that* $|c| = q \geqslant \left\lceil\frac{1}{\mu}\ln\left(\frac{4}{\varepsilon}\right)\right\rceil$ *is trivial.*

Proof. When the target is composed by $q \geqslant \left\lceil\frac{1}{\mu}\ln\left(\frac{4}{\varepsilon}\right)\right\rceil$ variables, then $MU = \left(\prod_{i\in\mathfrak{M}} p_i\right)\cdot\left(\prod_{k\in\mathfrak{U}} p_k\right) = \prod_{j\in\mathfrak{M}\cup\mathfrak{U}} p_j \leqslant (1-\mu)^q \leqslant e^{-\mu q} \leqslant e^{-\mu\frac{1}{\mu}\ln\left(\frac{4}{\varepsilon}\right)} = \frac{\varepsilon}{4}$. Hence, for any h, $\mathbf{Cov}\left[h,c\right] \leqslant 4\cdot\frac{\varepsilon}{4}\cdot R\cdot(1-M) = \varepsilon\cdot R\cdot(1-M) \leqslant \varepsilon$. □

We want to use (3.22) to examine the difference $\Delta = \mathbf{Cov}\left[h', c\right] - \mathbf{Cov}\left[h,c\right].$ Before we do that though, we will need the following three lemmas that will be useful for giving us lower bounds on the non-zero values of the difference Δ.

Lemma 3.5.3. *Let* $x, y \in \mathcal{P}_n$ *(μ non-degenerate); i.e.* $x, y \in [\mu, 1-\mu]$. *Then, for the non-zero values of* $f(x,y) = 1 - \frac{x}{y}$ *it holds* $|f(x,y)| = \left|1-\frac{x}{y}\right| > \mu$.

Proof. We have $f(x,y) = \frac{y-x}{y}$. For $x \neq y$ (so that $f(x,y) \neq 0$), we have $|f(x,y)| = \left|\frac{y-x}{y}\right| = \frac{|y-x|}{|y|}$. The fraction takes its minimum non-zero value when $|y-x|$ is minimum and $|y|$ is maximum. Hence we have $|f(x,y)| \geqslant \frac{\mu}{1-\mu} > \mu$. □

Lemma 3.5.4. *Let* $x, y \in \mathcal{P}_n$ *(μ non-degenerate); i.e.* $x, y \in [\mu, 1-\mu]$, $c \in (0, y]$, *and* $f(x,y) = x - 1 + c\cdot(1-x/y)$. *Then* $|f(x,y)| \geqslant \mu$.

Proof. First of all, note that the analysis of Section 3.5 guarantees that $f(x,y) < 0 \ \forall(x,y) \in [\mu, 1-\mu] \times [\mu, 1-\mu]$. Hence, the minimum non-zero value of $|f(x,y)|$ is obtained either in a local optimum in the interior, or on the boundary of the domain. We have $\frac{\partial f}{\partial x} = 1 - \frac{c}{y}$ and $\frac{\partial f}{\partial y} = \frac{cx}{y^2}$. Hence, a candidate solution for a local optimum is the point $(0,c)$ where the partial derivatives vanish. However, $(0,c) \notin [\mu, 1-\mu] \times [\mu, 1-\mu]$. So, we have to examine the boundary only. We have:

$x = \mu$: Then, $f(\mu, y) = \mu - 1 + c(1 - \mu/y)$, with $\partial f/\partial y = cm/y^2 > 0$. Hence, $f(\mu, y)$ achieves its maximum (and as a consequence $|f(m,y)|$ its minimum) when $y = 1 - m$. We have $f(\mu, 1-\mu) = \mu - 1 + c(1 - \mu/(1-\mu)) = \mu - 1 + c \cdot \left(\frac{1-2\mu}{1-\mu}\right) \leqslant \mu - 1 + (1-\mu) \cdot \frac{1-2\mu}{1-\mu} = \mu - 1 + 1 - 2\mu = -\mu$.

$x = 1-\mu$: Then, $f(1-\mu, y) = 1 - \mu - 1 + c(1 - (1-\mu)/y)$. Again $\partial f/\partial y > 0$, hence, $f(1-\mu, 1-\mu) = -\mu$ is the maximum value of f along this boundary.

$y = \mu$: Then, $f(x, \mu) = x - 1 + c(1 - x/\mu)$. We have $\partial f/\partial x = 1 - c/\mu \geqslant 0$. If $c = \mu$, then $f(x,\mu) = -1 + \mu \leqslant -\mu$. If $c < \mu$, then $\partial f/\partial x > 0$, so we look at $f(1-\mu, \mu)$, for which it holds $f(1-\mu, \mu) = -\mu - c\frac{1-2\mu}{\mu} \leqslant -\mu$.

$y = 1-\mu$: Then, $f(x, 1-\mu) = x - 1 + c(1 - x/(1-\mu))$, with $\partial f/\partial x = 1 - c/(1-\mu)$. If $c = 1-\mu$, then $f(x, 1-\mu) = -\mu$, while if $c < 1-\mu$, then $\partial f/\partial x > 0$, and hence the maximum is obtained for $x = 1-\mu$, which is $f(1-\mu, 1-\mu) = -\mu$.

So, in every case we have that $f(x,y) \leqslant -\mu$, $\forall(x,y) \in [\mu, 1-\mu] \times [\mu, 1-\mu]$. □

Lemma 3.5.5. *Let* $x, y \in \mathcal{P}_n$ *(μ non-degenerate); i.e.* $x, y \in [\mu, 1-\mu]$, $c \in (0,1]$, *and* $f(x,y) = 1/x - 1 + c \cdot (1 - y/x)$. *Then* $|f(x,y)| > \mu$.

Proof. Again, by the analysis of Section 3.5 it holds that $f(x,y) > 0 \ \forall(x,y) \in [\mu, 1-\mu] \times [\mu, 1-\mu]$. Hence, the minimum non-zero value of $|f(x,y)|$ is obtained either in a local optimum in the interior, or on the boundary of the domain. Like in the previous case, there is no point in the interior of the domain that can make both of the derivatives vanish simultaneously. Hence, we have to examine again only the boundary. We have:

$x = \mu$: Then, $f(\mu, y) = 1/\mu - 1 + c(1 - y/\mu)$, with $\partial f/\partial y = -c/\mu < 0$. Hence, the minimum is obtained for $y = 1-\mu$, where it holds $f(\mu, 1-\mu) = \frac{1-\mu}{\mu} - c \cdot \left(\frac{1-2\mu}{\mu}\right) \geqslant \frac{1-\mu}{\mu} - \frac{1-2\mu}{\mu} = 1 > \mu$.

$x = 1-\mu$: Then, $f(1-\mu, y) = 1/(1-\mu) - 1 + c(1 - y/(1-\mu))$, with $\partial f/\partial y = -c/(1-\mu) < 0$. Hence, the minimum is obtained for $y = 1-\mu$, where it holds $f(1-\mu, 1-\mu) = \frac{\mu}{1-\mu} > \mu$.

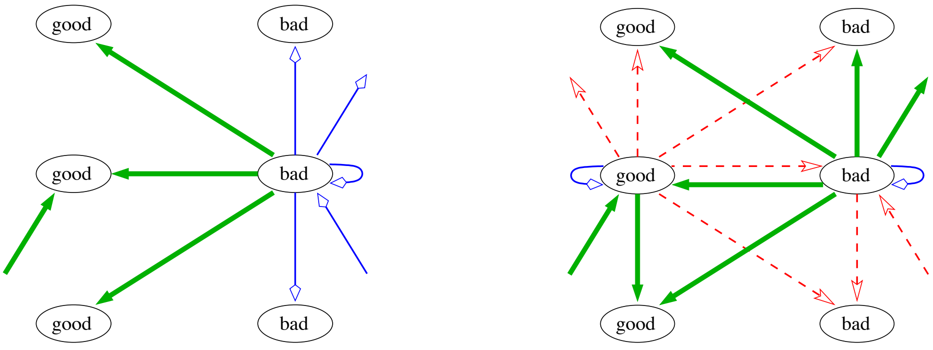
$y = \mu$: Then, $f(x, \mu) = 1/x - 1 + c(1 - \mu/x)$, with $\partial f/\partial x = -(1 - \mu c)/x^2 < 0$. Hence, we examine f for $x = 1-\mu$, where it holds $f(1-\mu, \mu) = \frac{\mu}{1-\mu} + c((1-2\mu)/(1-\mu)) \geqslant \mu/(1-\mu) > \mu$.

$y = 1-\mu$: Then, $f(x, 1-\mu) = 1/x - 1 + c(1 - (1-\mu)/x)$, with $\partial f/\partial x = -\frac{1-c(1-\mu)}{x^2} < 0$. Hence we examine f for $x = 1-\mu$, but we have already shown above that $f(1-\mu, 1-\mu) > \mu$.

So, in every case, $f(x,y) \geqslant \mu$, $\forall(x,y) \in [\mu, 1-\mu] \times [\mu, 1-\mu]$. □

We are now ready to proceed with the analysis of the difference Δ.

Comparing $h' \in N^+$ with h. We introduce a variable z in the hypothesis h. If z is good, then $\Delta = 4M^2RU(1-p_z) > 0$. If z is bad, then $\Delta = (p_z - 1)\,\mathbf{Cov}[h,c] \leqslant 0$. We have equality if $m = 0$; i.e. $M = 1$. **Lower Bound:** When adding a good variable we have $\Delta = 4W_hW_c(1-p_z) \geqslant 4\mu^{2q}$, since $|h| \leqslant q-1$. When adding a bad variable we have $|\Delta| = 4MRU(1-M)(1-p_z) \geqslant 4(MR)(MU)(1-M)(1-p_z) = 4W_hW_c(1-M)(1-p_z) \geqslant 4\mu^{q-1}\mu^q\mu \cdot \mu \geqslant 4\mu^{2q+1}$.



(a) $M = 1$ (b) $M < 1$

Figure 3.9: The style and the directions of arrows have the same interpretation as in the previous figures. Similarly, the hierarchy of nodes on levels has the same interpretation. Some arrows are missing in the left picture since there are no good variables in the hypothesis; i.e. $M = 1$.

Comparing $h' \in N^-$ with h. We remove a variable z from the hypothesis h. If z is good, then $\Delta = -4M^2RU(1/p_z - 1) < 0$. If z is bad, then $\Delta = (1/p_z - 1)\mathbf{Cov}[h, c] \geqslant 0$. We have equality if $m = 0$; i.e. $M = 1$. **Lower Bound:** When removing a good variable we have $|\Delta| \geqslant 4W_hW_c(1/p_z - 1) \geqslant 4\mu^{2q}(1/p_z - 1) \geqslant 4\mu^{2q}\frac{\mu}{1-\mu} > 4\mu^{2q+1}$. When removing a bad variable we have $\Delta = 4MRU(1-M)(1/p_z - 1) \geqslant 4(MR)(MU)(1-M)(1/p_z - 1) \geqslant 4\mu^q\mu^q\mu\frac{\mu}{1-\mu} > 4\mu^{2q+2}$. **Upper Bound:** Note, that it holds $|h| \geqslant q + 1$. When we remove a good variable, then $|\Delta| = 4W_hW_c(1/p_z - 1) \leqslant 4e^{-\mu(q+1)}(M/p_z)U(1 - p_z) \leqslant 4e^{-\mu(q+1)}e^{-\mu} = 4e^{-\mu(q+2)}$. When we remove a bad variable, then $\Delta = (1/p_z - 1)4MRU(1-M) = 4(1-p_z)M(R/p_z)U(1-M) \leqslant 4e^{-\mu}e^{-\mu q} \cdot 1 \cdot 1 = 4e^{-\mu(q+1)} \leqslant 4e^{-\mu}e^{-\mu\frac{1}{\mu}\ln\frac{4}{\varepsilon}} = 4 \cdot \frac{\varepsilon}{4} = \varepsilon$.

Comparing $h' \in N^{+-}$ with h. We swap a variable **out** with a variable **in.**

If **out** is good and **in** is good, then $\Delta = 4M^2RU(1 - p_{\mathbf{in}}/p_{\mathbf{out}})$.

If $p_{\mathbf{out}} \leqslant p_{\mathbf{in}}$, then $\Delta \leqslant 0$, with $\Delta = 0$ if $p_{\mathbf{out}} = p_{\mathbf{in}}$. If $p_{\mathbf{out}} > p_{\mathbf{in}} \Rightarrow \Delta > 0$.

If **out** is good and **in** is bad, then $\Delta = 4MRU \cdot ((p_{\mathbf{in}} - 1) + M \cdot (1 - p_{\mathbf{in}}/p_{\mathbf{out}}))$. We now examine the quantity $\kappa = (p_{\mathbf{in}} - 1) + M \cdot (1 - p_{\mathbf{in}}/p_{\mathbf{out}})$:

$p_{\mathbf{out}} \leqslant p_{\mathbf{in}}$: Then $(1 - p_{\mathbf{in}}/p_{\mathbf{out}}) \leqslant 0$, and $(p_{\mathbf{in}} - 1) < 0$. Therefore $\Delta < 0$.

$p_{\mathbf{out}} > p_{\mathbf{in}}$: Note $M \leqslant p_{\mathbf{out}}$. Hence, $\kappa < p_{\mathbf{in}} - 1 + 1 - p_{\mathbf{in}}/p_{\mathbf{out}} = p_{\mathbf{in}}(1 - 1/p_{\mathbf{out}}) < 0$.

If **out** is bad and **in** is bad, then $\Delta = (p_{\mathbf{in}}/p_{\mathbf{out}} - 1) \cdot \mathbf{Cov}[h, c]$ and $\mathbf{Cov}[h, c] \geqslant 0$:

$p_{\mathbf{out}} \leqslant p_{\mathbf{in}}$: In this case, $\Delta \geqslant 0$, and $\Delta = 0$ when $m = 0$, or $p_{\mathbf{out}} = p_{\mathbf{in}}$.

$p_{\mathbf{out}} > p_{\mathbf{in}}$: In this case $\Delta \leqslant 0$, and $\Delta = 0$ when $m = 0$.

If **out** is bad and **in** is good, then $\Delta = 4MRU(1/p_{\mathbf{out}} - 1 + M(1 - p_{\mathbf{in}}/p_{\mathbf{out}}))$. We examine the quantity $\kappa = 1/p_{\mathbf{out}} - 1 + M(1 - p_{\mathbf{in}}/p_{\mathbf{out}})$:

$p_{\mathbf{out}} < p_{\mathbf{in}}$: Note $M \leqslant 1$. Hence, $\kappa > 1/p_{\mathbf{out}} - 1 + 1 - p_{\mathbf{in}}/p_{\mathbf{out}} = (1 - p_{\mathbf{in}})/p_{\mathbf{out}} > 0$.

$p_{\mathbf{out}} \geqslant p_{\mathbf{in}}$: In this case $p_{\mathbf{in}}/p_{\mathbf{out}} \leqslant 1 \Rightarrow \kappa > 0 \Rightarrow \Delta > 0$.

Lower Bound: Swapping a good **out** with a good **in** yields $|\Delta| = 4M^2RU \cdot |1 - p_{\mathbf{in}}/p_{\mathbf{out}}| = 4W_hW_c \cdot |1 - p_{\mathbf{in}}/p_{\mathbf{out}}|$. So, by Lemma 3.5.3 we have $|\Delta| > 4\mu^q\mu^q\mu = 4\mu^{2q+1}$. Swapping a bad **out** with a bad **in** yields $|\Delta| = |p_{\mathbf{in}}/p_{\mathbf{out}} - 1| \cdot \mathbf{Cov}[h, c] \geqslant |p_{\mathbf{in}}/p_{\mathbf{out}} - 1| \cdot 4(MR)(MU)(1-M)$, and by Lemma 3.5.3 we get that $\Delta > \mu \cdot 4\mu^q\mu^q\mu = 4\mu^{2q+2}$. Swapping a good **out** with a bad **in** yields $\Delta = 4MRU \cdot ((p_{\mathbf{in}} - 1) + M \cdot (1 - p_{\mathbf{in}}/p_{\mathbf{out}}))$. Hence, by Lemma 3.5.4 we have $|\Delta| \geqslant 4(MR)(MU)\mu \geqslant 4\mu^q\mu^q\mu = 4\mu^{2q+1}$. Swapping a bad **out** with a good **in** yields $\Delta = 4MRU(1/p_{\mathbf{out}} - 1 + M(1 - p_{\mathbf{in}}/p_{\mathbf{out}}))$. Hence, by Lemma 3.5.5 we have $\Delta \geqslant 4(MR)(MU)\mu > 4\mu^q\mu^q\mu = 4\mu^{2q+1}$.

The effects of the different mutations are summarized in Figure 3.9. Compared to Figure 3.2, it is remarkably simple and uniform, and can be summarized as follows. If there are some mutual variables (i.e. good) in the hypothesis, then

- $\Delta > 0$ if a good variable is added, a bad variable is deleted, a bad variable is replaced by a good one, a good variable is replaced by a smaller good one, and a bad variable is replaced by a larger bad one,

- $\Delta < 0$ if a good variable is deleted, a bad variable is added, a good variable is replaced by a bad one, a good variable is replaced by a larger good one, and a bad variable is replaced by a smaller bad one,

- $\Delta = 0$ if a good variable is replaced by an equivalent good one, and a bad variable is replaced by an equivalent bad one.

If there are no mutual variables in the hypothesis, then

- $\Delta > 0$ if a good variable is added, or a good variable replaces a bad one.

- $\Delta = 0$ if a bad variable is added, deleted, or replaced by a bad one.

Note that the beneficiality or neutrality of a mutation is *not* determined by these observations; it also depends on the tolerances. Nevertheless, these properties are sufficient for an analogue of Theorem 3.3.3 on the structure of best approximations to hold for product distributions and the covariance fitness function.

Theorem 3.5.6 (Structure of Best Approximations)**.** *The best q-approximation of a target c, such that $|c| \geqslant 1$, is c itself if $|c| \leqslant q$, or any hypothesis formed by q good variables, such that the product $\prod_{i=1}^{q} p_i$ is minimized if $|c| > q$.*

As mentioned earlier, the existence of characterizations of best approximations is related to evolvability. This relationship is now illustrated for product distribution and the covariance fitness function. First we introduce an idealized version of evolution algorithms, where beneficiality depends on the precise value of the performance function.

Definition 3.5.7 (Unbounded-Precision Evolution Algorithm)**.** An evolution algorithm is unbounded-precision if, instead of (3.3) it uses

$$\begin{cases} \text{Bene} &= N \cap \left\{h' \mid \mathrm{Perf}_{\mathcal{D}_n}\left(h', c\right) > \mathrm{Perf}_{\mathcal{D}_n}\left(h, c\right)\right\} \\ \text{Neut} &= N \cap \left\{h' \mid \mathrm{Perf}_{\mathcal{D}_n}\left(h', c\right) = \mathrm{Perf}_{\mathcal{D}_n}\left(h, c\right)\right\} \end{cases}, \tag{3.23}$$

or, equivalently, arbitrary tolerance to determine which hypotheses are beneficial, neutral or deleterious. All other parts of the definition are unchanged.

Consider the following unbounded-precision evolution algorithm: starting from an arbitrary initial hypothesis, apply beneficial mutations as long as possible. Then beneficial mutations can add a good variable, delete a bad variable, replace a bad variable by a good one, replace a good variable by a smaller good one or replace a bad variable by a larger bad one. The amortized analysis argument of Theorem 3.5.9 in the next section shows that the number of steps is $O(n^2)$. Hence the following result holds.

Theorem 3.5.8. *The swapping algorithm using the covariance fitness function is an efficient evolution algorithm for monotone conjunctions over product distributions.*

3.5.1 Evolving Short Monotone Conjunctions under Product Distributions

The problem with applying the unbounded-precision algorithm to the bounded-precision model is that the presence of the U factor in Δ may make the performance differences superpolynomially small. If we assume that the product distribution is μ nondegenerate and the target is short then this cannot happen, and an analysis similar to Theorem 3.3.8 shows that we indeed get an efficient evolution algorithm. In Section 3.6 we give some remarks on possibilities for handling long targets. We set

$$q = \left\lceil \frac{1}{\mu} \ln \frac{4}{\varepsilon} \right\rceil \ .$$

Theorem 3.5.9. *Let $\mathcal{P}_n$ be a μ-nondegenerate product distribution. The swapping algorithm, using the covariance fitness function, evolves* non-empty *short ($1 \leqslant |c| \leqslant q$) monotone conjunctions starting from an initial hypothesis h_0 in $\mathcal{O}\left(nq + |h_0| \ln \frac{1}{\delta}\right)$ iterations, each iteration examining the performance of $\mathcal{O}\left(nq\right)$ hypotheses, and each performance being evaluated using sample size*

$$\mathcal{O}\left(\left(\frac{1}{\mu}\right)^4 \left(\frac{1}{\varepsilon}\right)^{(4/\mu)\ln(1/\mu)} \left(\ln n + \ln\frac{1}{\delta} + \ln\frac{1}{\mu} + \ln\frac{1}{\varepsilon}\right)\right).$$

Proof. The analysis of the proof is similar to that of Theorem 3.3.8.

Short Initial Hypothesis. Again, we are interested in the *non-zero* values of the quantity Δ so that, given representative samples, we can characterize precisely all the mutations. For the nonzero values of Δ we have:

$$|\Delta| \geqslant 4\mu^{2q+2}.$$

Therefore, we set the tolerance $t = 2\mu^{2q+2}$, and require accuracy for the empirical estimates $\epsilon = t = 2\mu^{2q+2}$.

Initially we want to exclude the case shown in Figure 3.9a; that is, form a hypothesis with at least one good variable. By selection of t, ϵ_M, all the arrows in both cases shown in Figure 3.9 are determined correctly. Therefore, if the algorithm is in state shown in Figure 3.9a, the set of beneficial mutations is composed either by adding a good variable or swapping a bad variable with a good one. In any case, in one step, the algorithm will reach the state shown in Figure 3.9b.

The algorithm is in the state shown in Figure 3.9b with a hypothesis $h_1 = \bigwedge_{i=1}^{m} x_i \wedge \bigwedge_{\ell=1}^{r} y_\ell$ such that $m \geqslant 1$ and $m + r \leqslant q$. Evolution will stop when there are no further beneficial mutations. We split the set of beneficial mutations to 2 further subsets. The first subset changes the quantities m, r and is composed by those mutations that introduce a good variable, remove a bad variable, or swap a bad variable with a good one. The second subset is composed by the other beneficial mutations which retain these values. Note that throughout the evolution the number of good variables is non-decreasing, and the number of bad variables is non-increasing.

Regarding mutations from the first subset, we add good variables or swap bad to good at most $|c| - m$ times since this is the amount of missing good variables. Moreover, removing a bad variable will happen at most $r = |h_1| - m$ times. Hence, at most $|c| + |h_1| - 2m$ mutations from the first subset can occur throughout the evolution.

We now look on the mutations from the second subset in discrete time frames of the evolution where the quantities m and r remain unchanged. In particular, regarding swaps that interchange bad with bad variables, we split the evolution in $r = |h_1| - m$ phases where the number of bad variables remain unchanged in each phase. During each such phase, consider the bad variables as members of a sorted array of size r. The algorithm will start with a configuration

a_1	a_2	$\cdots$	a_r
y_1	y_2	$\cdots$	y_r

such that $\mathbf{Pr}(y_1 = \text{TRUE}) \leqslant \mathbf{Pr}(y_2 = \text{TRUE}) \leqslant \ldots \leqslant \mathbf{Pr}(y_r = \text{TRUE})$, and will end the current phase with a configuration

a_1	a_2	$\cdots$	a_r
y_1'	y_2'	$\cdots$	y_r'

such that $\mathbf{Pr}(y_1' = \text{TRUE}) \leqslant \mathbf{Pr}(y_2' = \text{TRUE}) \leqslant \ldots \leqslant \mathbf{Pr}(y_r' = \text{TRUE})$, where in case of equalities order the variables with lexicographic ordering. We now look on the structure of the bad array throughout all r phases; see Table 3.1. Consider the entries of the arrays in a right to left fashion throughout all r phases. The crucial observation is that every entry a_i of the array, as long as it exists, follows a non-decreasing order throughout evolution. Hence, the entry a_r can take at most $n - |c| - r$ different values, the entry a_{r-1} can take at most $n - |c| - r - 1$ different values, and in general, the entry a_{r-i} can take at most $n - |c| - r - i$ different values, where $i \in \{0, 1, 2, \ldots, r-1\}$. Hence the number of swaps between bad variables can be at most

$$\sum_{i=0}^{r-1}(n - |c| - r - i).$$

We compute:

$$\begin{aligned}\sum_{i=0}^{r-1}(n - |c| - r - i) &= r(n - |c| - r) - \sum_{i=1}^{r-1} i \\ &= r(n - |c| - r) - \frac{r(r-1)}{2} \\ &= r\,(n - |c| - 3r/2 + 1/2)\end{aligned}$$

and the bound is tight for distributions where all the p_i's are distinct.

Finally, regarding swaps that interchange good with good variables, we follow a similar procedure. We split the evolution in $|c| - m$ phases where the number of good variables remain unchanged in each phase. This time we have the sequence of arrays shown in Table 3.2. We now look on the entries of the array in the same direction (right to left). This time, each entry a_i of the array follows an non-increasing order throughout evolution. Hence, entry $a_{|c|}$ can take at most 1 different values, $a_{|c|-1}$ can take at most 2 different values, and in general the entry a_i can take at most $|c|+1-i$ different values. Summing up we get

$$\sum_{i=1}^{|c|}(|c| + 1 - i).$$

In other words we have at most $\sum_{i=1}^{|c|} i = |c|(|c|+1)/2$ swaps, and the bound is tight for distributions where all the p_i's are distinct.

Complexity Analysis for Short Initial Hypothesis. Summing up over all the possibilities that were analyzed we obtain $1 + (|c| + |h_1| - 2m) + r(n - |c| - 3r/2 + 1/2) + |c|(|c|+1)/2 \leqslant 1 + (2q-2) + r(n - 1 - 0 + 1/2) + q(q+1)/2 \leqslant 2q - 1 + rn - r/2 + q^2/2 + q/2 \leqslant 2q - 1 + qn + q^2/2 + q/2 \leqslant qn + 2q^2$ steps in the worst case. In each of these steps, by Lemma 3.2.1, we have neighborhoods of size at most $|N| \leqslant 2qn$. Hence, for the entire process we have to estimate the performance of no more than $(qn + 2q^2) \cdot (2qn) = 2q^2n^2 + 4q^3n$ hypotheses, each one of them within accuracy $\epsilon = t = 2\mu^{2q+2}$. By the Hoeffding Bound (Proposition 2.2.10) when setting $\alpha = -1, \beta = 1, \epsilon = 2\mu^{2q+2}$, the performance of each hypothesis is not estimated within $\epsilon = 2\mu^{2q+2}$ of its true value with probability $e^{-2R\mu^{4q+4}}$. By the Union Bound (Proposition 2.2.5) the performance of each hypothesis is computed within $\epsilon = 2\mu^{2q+2}$ of its true value with probability at least $1 - \sum_{\text{all hypotheses}} e^{-2R\mu^{4q+4}} \geqslant 1 - (2q^2n^2 + 4q^3n) \cdot e^{-2R\mu^{4q+4}}$. We require now this probability to be at least $1 - \delta/2$ and hence it is enough if each empirical estimate is computed with at least $R \geqslant \left\lceil \frac{1}{2} \cdot \left(\frac{1}{\mu}\right)^{4q+4} \cdot \ln\left(\frac{4q^2n^2+8q^3n}{\delta}\right) \right\rceil$ samples; i.e. we require $\mathcal{O}\left(\left(\frac{1}{\mu}\right)^4 \cdot \left(\frac{1}{\varepsilon}\right)^{\frac{4}{\mu}\ln\frac{1}{\mu}} \cdot \left(\ln\frac{1}{\mu} + \ln\ln\frac{1}{\varepsilon} + \ln n + \ln\frac{1}{\delta}\right)\right)$ samples for the approximation of the performance of each hypothesis. Hence, the total number of samples for this phase is $\mathcal{O}\left((q^2n^2 + q^3n) \cdot \left(\frac{1}{\mu}\right)^4 \cdot \left(\frac{1}{\varepsilon}\right)^{\frac{4}{\mu}\ln\frac{1}{\mu}} \cdot \left(\ln\frac{1}{\mu} + \ln\ln\frac{1}{\varepsilon} + \ln n + \ln\frac{1}{\delta}\right)\right)$.

Long Initial Hypothesis. The arguments are similar to those in Theorem 3.3.8, and in $\mathcal{O}\left(|h_0| + \ln\frac{1}{\delta}\right)$ stages the algorithm forms a short hypothesis of size q. Then, we apply the analysis above.

In particular, regarding the shrinking process of the evolution, again by Lemma 2.2.15 we get that $t = \left\lceil \frac{8}{3} \cdot \left(|h_0| + \ln\left(\frac{4}{\delta}\right)\right) \right\rceil$ generations are enough in order to guarantee $|h_0|$ successes in the coin tossing process with probability at least $1 - \delta/4$. In other words, $\mathcal{O}\left(|h_0| + \ln\left(\frac{1}{\delta}\right)\right)$ generations are enough.

Similarly, we want to guarantee throughout the entire shrinking process all the empirical estimates of the performance of each hypothesis is within accuracy $\epsilon = 4e^{-\mu(q+1)}$ of their true values with probability at least $1 - \delta/4$. Again we use Proposition 2.2.10 with $\alpha = -1, \beta = 1, \epsilon = 4e^{-\mu(q+1)}$. The performance of any hypothesis is not computed within $\epsilon = 4e^{-\mu(q+1)}$ of its true value with probability $e^{-8Re^{-2\mu(q+1)}}$. The entire process lasts for $t = \left\lceil \frac{8}{3} \cdot \left(|h_0| + \ln\left(\frac{4}{\delta}\right)\right) \right\rceil$ steps, and in each step the neighborhood has size $|h_i|+1 \leqslant n+1$. By the Union Bound (Proposition 2.2.5) the probability that any empirical estimate is not within $\epsilon = 4e^{-\mu(q+1)}$ of its true value is at most $(n+1) \cdot t \cdot e^{-8Re^{-2\mu(q+1)}}$. We now require to bound this quantity from above by $\delta/4$ and hence we need $R \geqslant \left\lceil \frac{1}{8} \cdot e^{2\mu(q+1)} \ln\left(\frac{4(n+1)t}{\delta}\right) \right\rceil$ samples per empirical estimate computation. Noting that $e^{2\mu} \leqslant e$, as well as $e^{2\mu q} = e^{2\mu\left\lceil \frac{1}{\mu}\ln\frac{4}{\varepsilon} \right\rceil} \leqslant e^{2\mu(\frac{1}{\mu}\ln\frac{4}{\varepsilon}+1)} = e^{2\mu} \cdot \frac{16}{\varepsilon^2} \leqslant \frac{16e}{\varepsilon^2}$ it follows that $R \geqslant \left\lceil 2 \cdot \frac{e^2}{\varepsilon^2} \cdot \ln\left(\frac{4(n+1)t}{\delta}\right) \right\rceil$ samples are enough for every empirical estimate. In other words, we need $\mathcal{O}\left(\left(\frac{1}{\varepsilon}\right)^2 \left(\ln n + \ln|h_0| + \ln\frac{1}{\delta} + \ln\ln\frac{1}{\delta}\right)\right)$ samples for the approximation of the performance of each hypothesis. As a consequence, the total number of samples for this phase is $\mathcal{O}\left(n \cdot \left(|h_0| + \ln\frac{1}{\delta}\right) \cdot \left(\frac{1}{\varepsilon}\right)^2 \cdot \left(\ln n + \ln|h_0| + \ln\frac{1}{\delta} + \ln\ln\frac{1}{\delta}\right)\right)$. □

3.6 Further Remarks

It appears that from the perspective of learning theory, one of the remarkable features of Valiant's new model of evolvability is that it puts basic, well-understood learning problems in a new light and poses new questions about their learnability. One of these new questions is the performance of basic, simple evolution mechanisms, like the swapping algorithm for monotone conjunctions. The results of this chapter suggest that the analysis of these mechanisms may be an interesting challenge.

There seem to be many interesting directions of study for the future. For example, there is a similar swapping-type learning algorithm for decision lists, where a single step exchanges two tests in the list [124, 20]. Can such an algorithm be used in the evolution model? A positive answer could give an alternative to Michael's Fourier-based approach [94]. Another idea would be the study of an algorithm with an intuitive neighborhood on the Fourier spectrum, like in Michael's case [94], but for a different Boolean function.

Algorithm 1: The Mutator Function under the Uniform Distribution

```
Input: $q \in \mathbb{N}^*$, samples $s_{M,1}$, samples $s_{M,2}$, a hypothesis h, a target c.
Output: a new hypothesis $h'$
if $|h| > 0$ then Generate $N^-$ ;
else $N^- \leftarrow \emptyset$;
;
if $|h| < q$ then Generate $N^+$ ;
else $N^+ \leftarrow \emptyset$;
;
if $|h| \leqslant q$ then Generate $N^{+-}$ ;
else $N^{+-} \leftarrow \emptyset$;
;
$v_b \leftarrow$ GetPerformance($h$);
Initialize $Bene$, $Neutral$ to $\emptyset$;
if $|h| \leqslant q$ then $t \leftarrow 2^{-2q}$;
else $t \leftarrow 2^{1-q}$;                                   /* set tolerance */
;
for $x \in N^+, N^-, N^{+-}$ do
    SetWeight($x$, $h$, $N^+$, $N^-$, $N^{+-}$);
    if $|x| \leqslant q$ then SetPerformance($x$, $c$, $s_{M,1}$);   /* $s_{M,1}$ examples */
    ;
    else SetPerformance($x$, $c$, $s_{M,2}$);                    /* $s_{M,2}$ examples */
    ;
    $v_x \leftarrow$ GetPerformance($x$);
    if $v_x \geqslant v_b + t$ then $Bene \leftarrow Bene \cup \{x\}$;
    ;
    else if $v_x \geqslant v_b - t$ then $Neutral \leftarrow Neutral \cup \{x\}$;
    ;
SetWeight($h$, $h$, $N^+$, $N^-$, $N^{+-}$);
$Neutral \leftarrow Neutral \cup \{h\}$;
if $Bene \neq \emptyset$ then return RandomSelect($Bene$);
;
else return RandomSelect($Neutral$);
;
```

Table 3.1: The sequence of bad arrays throughout evolution.

	a_1	a_2	a_3	$\cdots$	a_{r-1}	a_r
Phase 1:	$y_{1,1}$	$y_{1,2}$	$y_{1,3}$	$\cdots$	$y_{1,r-1}$	$y_{1,r}$
	$y'_{1,1}$	$y'_{1,2}$	$y'_{1,3}$	$\cdots$	$y'_{1,r-1}$	$y'_{1,r}$
Phase 2:		$y_{2,1}$	$y_{2,2}$	$\cdots$	$y_{2,r-2}$	$y_{2,r-1}$
		$y'_{2,1}$	$y'_{2,2}$	$\cdots$	$y'_{2,r-2}$	$y'_{2,r-1}$
$\vdots$			$\ddots$	$\vdots$	$\vdots$	$\vdots$
				$\ddots$	$\vdots$	$\vdots$
Phase $r-1$:					$y_{r-1,1}$	$y_{r-1,2}$
					$y_{r-1,1}$	$y_{r-1,2}$
Phase ℓ:						$y_{r,1}$
						$y'_{r,1}$

Table 3.2: The sequence of good arrays throughout evolution.

	a_1	$\cdots$	a_m	a_{m+1}	$\cdots$	$a_{\lvert c\rvert}$
Phase 1:	$x_{1,1}$	$\cdots$	$x_{1,m}$			
	$x'_{1,1}$	$\cdots$	$x'_{1,m}$			
Phase 2:	$x_{2,1}$	$\cdots$	$x_{2,m}$	$x_{2,m+1}$		
	$x'_{2,1}$	$\cdots$	$x'_{2,m}$	$x'_{2,m+1}$		
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	
Phase $\lvert c\rvert - m$:	$x_{\lvert c\rvert-m,1}$	$\cdots$	$x_{\lvert c\rvert-m,m}$	$x_{\lvert c\rvert-m,m+1}$	$\cdots$	$x_{\lvert c\rvert-m,\lvert c\rvert}$
	$x'_{\lvert c\rvert-m,1}$	$\cdots$	$x'_{\lvert c\rvert-m,m}$	$x'_{\lvert c\rvert-m,m+1}$	$\cdots$	$x'_{\lvert c\rvert-m,\lvert c\rvert}$

Chapter 4

Multiple Instance Learning

Multiple-instance or multi-instance learning (MIL) is a variant of the standard PAC model of concept learning where, instead of receiving labeled instances as examples, the learner receives labeled bags, that is, labeled sets of instances. A bag is labeled positive if it contains at least one positive example, and it is labeled negative otherwise. Instances in a bag are usually assumed to be independent and identically distributed. This setting, introduced by Dietterich *et al.* [36], is natural for several learning applications, for example, in drug design and image classification. In drug design, a bag may consist of several shapes of a molecule and it is labeled positive if some shape binds to a specific binding site. In image classification, a bag may be a photo containing several objects and it is labeled positive if it contains some object of interest.

Blum and Kalai [15] showed that every learning problem that is efficiently learnable with statistical queries is also efficiently learnable in the MIL model, and, more generally, the same holds for problems efficiently learnable with one-sided random classification noise. This implies the efficient multi-instance learnability of all known efficiently PAC-learnable classes. A detailed study of sample sizes in the MIL model was initiated Sabato and Tishby [114]. They proved a general upper bound for the VC dimension of bags, and a lower bound for the concept class of halfspaces. Kundakcioglu *et al.* [81] considered margin maximization for bags of halfspaces and gave NP-completeness and experimental results.

In this note we continue the study of multi-instance learning of halfspaces. We improve the VC dimension lower bound of [114] from $\Omega(\log r)$ to $\Omega(d \log r)$, where d is the dimension and r is the bag size, which is optimal up to order of magnitude. We also show that the same lower bound holds for bags over every sufficiently large point set in general position. Thus the situation is somewhat analogous to standard halfspaces, where every simplex forms a maximum shattered set. The proofs are based on cyclic polytopes. We also show that hypothesis finding for bags of halfspaces is NP-complete, using a variant of the construction of [81]. These two results, in view of the well-known relationship between PAC-learnability, VC dimension and hypothesis finding, indicate differences between the PAC and MIL-PAC models.

Active learning is another variant of PAC learning. In this model the learner can decide whether to request the label of a random instance, and the complexity of an algorithm is measured by the number of label requests (see, e.g., Dasgupta [31]). Multi-instance active learning (MIAL) has been proposed by Settles *et al.* [121] and has been studied in several machine learning papers. We observe that the general active learning results of Hanneke [62] and Friedman [49] apply to the multi-instance setting as well.

There are several open problems related to the multi-instance learning of halfspaces. Some of these are discussed in the concluding section of the chapter.

A halfspace in $\mathbb{R}^d$ is given as $H = \{x \in \mathbb{R}^d : w \cdot x \geqslant t\}$, for weight vector $w \in \mathbb{R}^d$ and threshold $t \in \mathbb{R}$. A bag of size r, or an r-bag, is an r-element multiset $B = \{x_1, \ldots, x_r\}$ of $\mathbb{R}^d$. An r-bag B is positive for H if $B \cap H \neq \emptyset$, and B is negative for H otherwise. A set of bags $\mathcal{B} = \{B_1, \dots B_s\}$ is shattered by halfspaces if for every ± labeling of the bags there is halfspace that assigns the same labels to the bags in $\mathcal{B}$. The VC dimension of r-bags for d-dimensional halfspaces is the largest s such that there are s shattered bags. For $r = 1$ one gets the usual notion of VC dimension of halfspaces and it is a basic fact that this equals $d + 1$.

4.1 The VC Dimension of r-Bags for d-Dimensional Halfspaces

Sabato and Tishby [114] showed that the VC dimension of r-bags for any concept class is essentially at most a $\log r$ factor larger than the VC dimension of the concept class. We formulate their result in a slightly different form.

Theorem 4.1.1 ([114]). *For any concept class of VC dimension $\tilde{d}$, the VC dimension of r-bags is $O(\tilde{d} \log r)$.*

Proof. Let $\mathcal{B} = \{B_1, \dots B_s\}$ be a shattered set of r-bags. Then $\mathcal{B}$ contains at most $r \cdot s$ instances, and by Proposition 2.5.2 those can be classified by concepts in the class in at most $((ers)/\tilde{d})^{\tilde{d}}$ many ways. The classification of the instances in the bag determines the classification of the bags. Thus

$$2^s \leqslant \left(\frac{ers}{\tilde{d}}\right)^{\tilde{d}}.$$

Writing $x = s/\tilde{d}$ this becomes $2^x/x \leqslant er$. The function $2^x/x$ is monotone if $x \geqslant 1/\ln 2$. Thus it is sufficient to show that $2^x/x > er$ for $x = \log r + 2 \log\log r$, if r is sufficiently large, which follows directly. □

Sabato and Tishby showed that the VC dimension of r-bags of halfpaces in the plane is at least $\lfloor \log r \rfloor + 1$, which implies the same bound for higher dimensions. We now prove a lower bound by adding the 'missing' factor d, which is optimal in order of magnitude by Theorem 4.1.1.

Theorem 4.1.2. *The VC dimension of d-dimensional halfspaces over bags of size r is at least $\lfloor d/2 \rfloor (\lfloor \log r \rfloor + 1)$.*

Proof. Let $\ell \in \mathbb{N}$ and consider $n = \lfloor d/2 \rfloor \cdot 2^{\ell+1}$ points on the moment curve. Let $t_1 < \cdots < t_n$ be arbitrary and consider the set of n instances $X = \{x(t_1), \dots, x(t_n)\}$. Divide X into $\lfloor d/2 \rfloor$ blocks of size $2^{\ell+1}$ each, as this is shown in Figure 4.1, i.e., let

$$X_i = \{x(t_j) \, : \, (i-1) \cdot 2^{\ell+1} < j \leqslant i \cdot 2^{\ell+1}\}, \; i = 1, \dots, \left\lfloor \frac{d}{2} \right\rfloor.$$

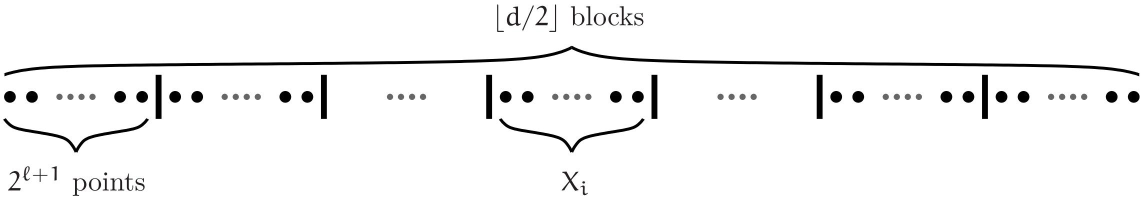


Figure 4.1: Start with $n = \lfloor \frac{d}{2} \rfloor \cdot 2^{\ell+1}$ points on the moment curve, in $\lfloor \frac{d}{2} \rfloor$ blocks of size $2^{\ell+1}$ each.

Let $f_i : X_i \rightarrowtail\!\!\!\!\rightarrow 2^{\{(i-1)\cdot(\ell+1)+1,\ i\cdot(\ell+1)\}}$ be a bijection between X_i and the powerset of $\{(i-1)\cdot(\ell+1)+1,\ i\cdot(\ell+1)\}$, and create $s = \lfloor d/2 \rfloor \cdot (\ell+1)$ bags so that

$$B_k = \{x(t_j) \,:\, k \in f_i(x(t_j))\}$$

for every k such that $(i-1)\cdot(\ell+1) < k \leqslant i\cdot(\ell+1)\}$. We claim that $\{B_1, \ldots, B_s\}$, with $s = \lfloor d/2 \rfloor \cdot (\ell+1)$, is a family of bags of size $r = 2^\ell$ shattered by d-dimensional halfspaces. Each bag is of size $r = 2^\ell$ as it contains a half of a block. For any subset of the bags $S \subseteq \{1, \ldots, s\}$ let $S_i = S \cap \{(i-1)\cdot(\ell+1)+1,\ i\cdot(\ell+1)\}$ and let $x(t_{j(i)})$ be the point such that $f_i(x(t_{j(i)})) = S_i$, for $i = 1, \ldots, \lfloor d/2 \rfloor$. Then the set $\{x(t_{j(i)}) \,:\, i = 1, \ldots, \lfloor d/2 \rfloor\}$ can be separated from the rest of X by a halfspace, and that halfspace classifies precisely those bags B_k as positive for which $k \in S$. Thus the family of bags is indeed shattered by halfspaces. The VC dimension bound follows directly from the definition of s and r. □

Now we prove a strengthening of Theorem 4.1.2. A finite subset of $\mathbb{R}^d$ is in *general position* if all its $(d+1)$-subsets are affinely independent, i.e., have no linear combination equal to 0, with coefficients adding up to 0. Halfspaces in $\mathbb{R}^d$ shatter *every* simplex, i.e., every set of $(d+1)$ points in general position. In analogy to this fact, we prove a VC dimension lower bound similar to Theorem 4.1.2 for bags of halfspaces when the instances are restricted to *any* sufficiently large subset in general position. The proof uses another property of cyclic polytopes. The following lemma is referred to as "unpublished 'folklore' " and proven in an oriented matroid version by Cordovil and Duchet [26] [1]. It is also given as an exercise in Matousek [90]. Again, we give a simple proof for completeness.

Lemma 4.1.3. *(See [26, 90].) There is a function* $f(d, n)$ *such that every set* A *of* $m \geqslant f(d, n)$ *points in general position in* $\mathbb{R}^d$ *contains* $n \geqslant d+1$ *points such that their convex hull has the same structure as a* d*-dimensional cyclic polytope on* n *vertices.*

Proof. The orientation of a d-dimensional ordered simplex $(a_0, \ldots, a_d)$ is the sign (+ or -) of the determinant with columns $a_1 - a_0, \ldots, a_d - a_0$, or, equivalently, with columns $a'_0, \ldots, a'_d$, where the primes denote an added first component of 1 to each vector.

Consider a d-dimensional cyclic polytope and let $x(t_{i_1}), \ldots, x(t_{i_{d+1}})$ be $d+1$ vertices of the polytope. The orientation of the simplex formed by these points using the increasing ordering of the parameters is +, as the corresponding determinant is a Vandermonde determinant.

Put $f(d, n) = R(d+1, n)$ in Proposition 2.4.1 and consider a set A of $m \geqslant f(d, n)$ points in general position. Fix an arbitrary ordering $<$ of the elements of A. Color each $(d+1)$-subset of A with the orientation (+ or −) of the corresponding simplex, ordered according to the fixed ordering. Note that the determinant that gives the orientation of every simplex is always non-zero, as these are points in general position. Then there is a subset $\{a_1, \ldots, a_n\}$ of A such that all ordered simplices from that subset have the same orientation.

Consider an arbitrary ordered d-subset $v_1 < \ldots < v_d$ of A. Denote by H the hyperplane determined by these points. Then for any other point $v \in A$, the orientation of the ordered simplex $(v, v_1, \ldots, v_d)$ determines which side of H contains v. Thus vertices $v_1, \ldots, v_d$ form a facet if and only if the sign of the determinant $\mathrm{det}(v', v'_1, \ldots, v'_d)$ is the same for every vertex v. This, however, is the same as Gale's evenness condition. Thus the face structure of the convex hull of $\{a_1, \ldots, a_n\}$ is the same as that of a cyclic polytope on n vertices. □

Theorem 4.1.4. *There is a function* $g(d, r)$ *such that for every set* A *of* $m \geqslant g(d, r)$ *points in general position in* $\mathbb{R}^d$*, halfspaces over bags of size* r *from* A *have VC dimension at least* $\lfloor d/2 \rfloor(\log r + 1)$.

Proof. The result follows by combining the construction of Theorem 4.1.2 with Lemma 4.1.3, setting $g(d, r) = f(d, dr)$. □

[1] The paper is an updated version of an unpublished, but circulated, manuscript from 1986/87.

4.2 NP-Completeness of Hypothesis Finding

The hypothesis-finding problem for r-bags for d-dimensional halfspaces is the following: given a set of labeled r-bags in $\mathbb{R}^d$, is there a halfspace that assigns these labels to the bags? The reduction below is a variant of a reduction in Kundakcіouglu *et al.* [81].

Theorem 4.2.1. *The hypothesis finding problem for r-bags of d-dimensional halfspaces is NP-complete for every fixed $r \geqslant 3$.*

Proof. We give a reduction from 3-SAT (containment in NP is trivial). Let $C_1, \ldots, C_m$ be an instance of 3-SAT over variables $x_1, \ldots, x_d$. Let e_i be the i'th unit vector in $\mathbb{R}^d$. For $j = 1, \ldots, m$ let B_j be a positive bag containing e_i if x_i is in C_j, and $-e_i$ if $\neg x_i$ is in C_j. For $i = 1, \ldots, d$ let B'_i be a positive bag containing e_i and $-e_i$. Finally, let B^* be a negative bag containing 0. We claim that the original formula is satisfiable iff the there is a consistent hypothesis for the set of bags described.

Let $(a_1, \ldots, a_d)$ be a satisfying truth assignment. Then the halfspace $w_1u_1 + \ldots + w_du_d \geqslant 1$ is consistent, where $w_i = 1$ if $a_i = 1$ and $w_i = -1$ otherwise, for $i = 1, \ldots, d$.

In the other direction, let $w_1u_1 + \ldots + w_du_d \geqslant t$ be a consistent hypothesis. Then $t > 0$ as B^* is negative. Also, $w_i \neq 0$, as B'_i is positive. It follows directly that the truth assignment defined by $a_i = sign(w_i)$ satisfies the formula. □

Note that this construction uses bags of size at most 3 (or r in the general case). Adding points to the bags sufficiently close to the given ones and slightly modifying the threshold one can get the same result for bags of the same size.

4.3 Further Remarks and Open Problems

We showed that the VC dimension of r-bags of d-dimensional halfspaces is $\Theta(d \log r)$ over every sufficiently large point set in general position and hypothesis finding for r-bags of d-dimensional halfspaces is NP-complete. This means that, unlike the case of learning halfspaces, one does not get an efficient PAC learning algorithm by drawing $O(d \log r)$ random bags and finding a consistent hypothesis. On the other hand, the result of Blum and Kalai [15] *does* provide an efficient algorithm with sample size polynomial in r and d.

This raises two open questions. What is the minimal sample size of r-bags sufficient for efficiently learning d-dimensional halfspaces? What is the minimal sample size of r-bags for PAC learning d-dimensional halfspaces without taking computational complexity into account? For the second question note that distributions over bags generated from arbitrary distributions over instances form a subclass of all possible distributions over bags[2], thus the VC dimension only provides an upper bound. Multi-instance learning under more general settings has been discussed by Auer *et al.* [7] and by Sabato and Tishby [114].

Further discussion related to MIL will be presented in Chapter 7 because we need some notions from active learning that are introduced in Chapter 5.

[2] This explains why, unlike the standard setting, the efficient PAC learning algorithm of Blum and Kalai [15] does not lead to an efficient hypothesis finding algorithm for bags.

Chapter 5

A Remark on Active Learning of Monotone Conjunctions

Active learning (AL) is another variant of the standard PAC model of concept learning. The aim of AL is to reduce the number of queries needed in order to learn a target concept. This is achieved by allowing the learner to choose the samples from which it learns. There are three main settings under which AL is performed.

Membership Query Synthesis. In the membership queries framework [4] the learner requests the labels of any data point in the input space. In this framework there is no underlying distribution, but rather the learner generates the instances for which the label will be requested. However, in real-world applications this approach may result in strange situations, since the automated algorithm may generate instances that no expert oracle can classify. An example illustrating this problem is given by Lang and Baum in [82] where they encountered query images generated by the learner that contained only artificial symbols with no natural meaning as characters.

Stream-Based Selective Sampling. In stream-based selective sampling [6, 24] a sequence of data points from the input space is presented to the learner and at each step the learner decides whether to query the label of the specific instance or not. There are different ways by which one can decide whether to query a specific instance or not. For example one approach is to query the more informative instances with higher probability [29]. Another example is to compute a region of uncertainty and only query the points that fall within that region [24]. In this last example a natural approach is to consider the version space [95, 96] of the target concept class and only query the points that can be labeled both ways among the remaining hypotheses in the version space. Real-world problem domains where stream-based selective sampling has been studied include part-of-speech tagging [29], sensor scheduling [80], learning ranking functions for information retrieval [146], and word sense disambiguation [50].

Pool-Based Sampling. In pool-based sampling [86], the learner is given access to a large pool of unlabeled data points and the aim is to choose only a fraction of those points in order to deduce the target concept. Apart from text classification [86, 92, 133, 70] pool-based sampling has been studied in other real-world problem domains such as information extraction [131, 120], image classification and retrieval [132, 147], video classification and retrieval [145, 65], speech recognition [134], and cancer diagnosis [88] to name a few. Note that the main difference between pool-based AL and stream-based AL is that pool-based AL evaluates and ranks the whole collection of the available examples and decides which is the best query, rather than taking this decision online and decide whether to query an individual example or not as it is done in the stream-based case.

Typically, there is a further distinction for each problem the learner is facing. This distinction refers to the *separable* versus the *non-separable* case. In the separable case we assume that the training points can be classified correctly by our hypothesis space $\mathcal{H}$, while in the non-separable case we do not assume that all of the training points can be classified correctly by our hypothesis space $\mathcal{H}$. Note that this distinction has similar flavor to *proper* learning versus *agnostic* learning [75]. Related work in agnostic active learning includes [32, 9, 60].

5.1 The Mellow Algorithm

Cohn, Atlas, and Ladner introduced in [24] an algorithm for actively learning separable data; see also [31]. This algorithm is referred to as the *mellow algorithm* or simply *CAL* due to the initials of its authors. The mellow algorithm is constantly maintaining the version space that is consistent with all the training examples seen so far. Hence, at time step $t \geqslant 1$ the algorithm maintains the version space $\mathcal{H}_t \subseteq \mathcal{H}$. When the new training point x_t is presented, the mellow algorithm queries this point only if there are hypotheses $h_1, h_2 \in \mathcal{H}_{t-1}$ such that $h_1(x_t) \neq h_2(x_t)$, otherwise, all the hypotheses in the version space agree about the label y_t of x_t and hence no queries are needed. Algorithm 2 has the details.

Algorithm 2: The Mellow Algorithm

```
Input: The hypothesis space $\mathcal{H}$.
Output: At time step $t$, the version space $\mathcal{H}_t$ that is consistent with all the training examples
        $x_1, x_2, \ldots, x_t$.
$t \leftarrow 0$;
$\mathcal{H}_0 \leftarrow \mathcal{H}$;
while TRUE do
    $t \leftarrow t + 1$;
    Receive unlabeled data point $x_t$;
    if disagreement in $\mathcal{H}_{t-1}$ about $x_t$'s label then
        query label $y_t$ of $x_t$;
        $\mathcal{H}_t \leftarrow \{h \in \mathcal{H}_{t-1} : h(x_t) = y_t\}$;
    else
        $\mathcal{H}_t \leftarrow \mathcal{H}_{t-1}$;
```

In fact, one does not necessarily need to explicitly maintain the entire version space $\mathcal{H}_t$, since maintaining the training points that have been presented together with their, possibly inferred, labels, implicitly gives the most general consistent version space with these training points. Algorithm 3 has the details.

Algorithms 2 and 3 were taken from [31].

5.2 Disagreement Coefficient

Below we describe the combinatorial notion of the disagreement coefficient which was introduced by Hanneke in [62].

Definition 5.2.1 (Ball of Radius ε)**.** For a target concept c, a ball $\mathbf{B}_{\mathcal{D}}(c, \varepsilon)$ of radius ε is defined to be the set of all the hypotheses which have error at most ε with respect to the target c under the distribution $\mathcal{D}$.

For the uniform distribution; that is, $\mathcal{D} = \mathcal{U}$, we will write $\mathbf{B}_{\mathcal{U}}(c, \varepsilon)$.

Definition 5.2.2 (Disagreement Region)**.** For a target concept c and a ball $\mathbf{B}_{\mathcal{D}}(c, \varepsilon)$, the disagreement region is defined to be the set of all the instances that can be classified in more than one way by the hypotheses found in the ball $\mathbf{B}_{\mathcal{D}}(c, \varepsilon)$.

Algorithm 3: The Mellow Algorithm without explicitly maintaining the version space $\mathcal{H}_t$

```
Input: $\emptyset$
Output: At time step $t$, the set $S$ of training points together with their labels
$t \leftarrow 0$;
$S \leftarrow \emptyset$;                                        /* points seen so far */
while TRUE do
    $t \leftarrow t + 1$;
    Receive unlabeled data point $x_t$;
    if learn($S \cup (x_t, -1)$) and learn($S \cup (x_t, +1)$) both return an answer then
        query label $y_t$ of $x_t$;
    else
        set $y_t$ to whichever label succeeded;
    $S \leftarrow S \cup (x_t, y_t)$;
```

Definition 5.2.3 (Disagreement Coefficient [62, 31])**.** The disagreement coefficient is defined to be

$$\rho_{c,\mathcal{D}_n} = \sup_{\varepsilon > 0} \frac{\mathbf{Pr}\left(\mathrm{DIS}\left(\mathbf{B}_{\mathcal{D}}\left(c, \varepsilon\right)\right)\right)}{\varepsilon} .$$

5.2.1 The Mellow Algorithm and the Disagreement Coefficient

We follow the notation in [31]. Let $\mathcal{L}_{\text{CAL}}\left(\varepsilon, \delta\right)$ be the smallest integer t_0 such that for all $t \geqslant t_0$ the probability that some hypothesis $h \in \mathcal{H}_t$ has error more than ε is less than or equal to δ. Since the dependence of $\mathcal{L}_{\text{CAL}}\left(\varepsilon, \delta\right)$ upon δ is modest, at most polylog$(1/\delta)$, the following theorem ignores δ and speaks only of $\mathcal{L}_{\text{CAL}}\left(\varepsilon\right)$.

Theorem 5.2.4 (Label Complexity of Mellow Algorithm; [62])**.** *Suppose $\mathcal{H}$ has finite VC dimension d, and the learning problem is separable, with disagreement coefficient ρ. Then,*

$$\mathcal{L}_{\text{CAL}}\left(\varepsilon\right) \leqslant \widetilde{\mathcal{O}}\left(\rho \cdot d \cdot \log \frac{1}{\varepsilon}\right) ,$$

where the $\widetilde{\mathcal{O}}\left(\cdot\right)$ notation suppresses terms logarithmic in ρ, d, and $\log(1/\varepsilon)$.

The important point of Theorem 5.2.4 is that while a typical supervised learner would need $\Omega\left(d/\varepsilon\right)$ examples to achieve error less than ε with high probability (see Theorem 2.5.4), in the active learning setting, this particular theorem has linear dependence with respect to the disagreement coefficient ρ and hence when the disagreement coefficient has at most logarithmic dependence on ε, Hanneke's bound implies that the mellow algorithm achieves an exponential speedup compared to traditional supervised learning. Moreover, quoting Dasgupta [31], this is done *"without any effort at finding maximally informative points!"*

5.2.2 Summary of Results for the Disagreement Coefficient

In Section 5.3 we compute bounds for the disagreement coefficient of monotone conjunctions under the uniform distribution $\mathcal{U}_n$ by studying the disagreement region.

- We compute the exact value for the empty target and the target of size 1. This is done in Theorems 5.3.1 and 5.3.2 respectively.

- For targets of size k such that $2 \leqslant k \leqslant n-1$ we give a general upper bound of $\mathcal{O}\left(2^k\right)$. This is done in Theorem 5.3.14.
- For targets of size k such that $2 \leqslant k \leqslant \lfloor n/2 \rfloor$ we show that the disagreement coefficient of these targets is $\Theta\left(2^k\right)$. This is done in Theorem 5.3.16.
- A lower bound of $\Omega\left(\frac{1}{n} \cdot 2^{(H(1-k/n)-(1-k/n))n}\right)$ is given for targets of size k such that $\lfloor n/2 \rfloor + 1 \leqslant k \leqslant 2n/3$. For targets of size $2n/3 < k \leqslant n-2$ we give a lower bound of $\Omega\left(\frac{1}{n} \cdot \left(\frac{3}{2}\right)^n\right)$. Both of these bounds are given in Lemma 5.3.17.
- An upper bound of $\mathcal{O}\left(2^{(H(1-k/n)-(1-k/n))n}\right)$ is given for targets of size k such that $\lfloor n/2 \rfloor + 2 \leqslant k \leqslant 2n/3$. For targets of size $2n/3 < k \leqslant n-2$ we give an upper bound of $\mathcal{O}\left(\left(\frac{3}{2}\right)^n\right)$. Both of these bounds are given in Lemma 5.3.18.
- Theorem 5.3.19 gives a summary of the bounds for targets of size k such that $\lfloor n/2 \rfloor + 1 \leqslant k \leqslant n-2$. Note that the upper bound for the case where $k = \lfloor n/2 \rfloor + 1$ is not given by Lemma 5.3.18, but instead from Theorem 5.3.14.
- For targets of size $n-1$ and n we give in both cases lower bounds of $\Omega\left(\frac{1}{n} \cdot \left(\frac{3}{2}\right)^n\right)$ and upper bounds of $\mathcal{O}\left(\left(\frac{3}{2}\right)^n\right)$. This is done in Theorems 5.3.13 and 5.3.6 respectively.

5.3 Disagreement Coefficient for Monotone Conjunctions under $\mathcal{U}_n$

Here we examine the disagreement coefficient of monotone conjunctions under the uniform distribution $\mathcal{U}_n$. Apart from the very recent paper of Balcan, Berlind, Ehrlich, and Liang [8], to the best of our knowledge, Boolean functions have not been studied in the framework of AL. In the case of targets of size $k \geqslant \lfloor n/2 \rfloor + 2$ we are going to make extensive use of Proposition 2.2.2 in order to upper bound sums of binomial coefficients. Similarly, in the case of targets of size $k \geqslant \lfloor n/2 \rfloor + 1$, for lower bounding sums of binomial coefficients we are going to use the lower bound of Proposition 2.2.1 for the biggest binomial coefficient that appears in each sum and use this bound for the entire sum every time.

Given a target conjunction c and a hypothesis conjunction h, the probability of the error region of h with respect to c can be found by counting truth assignments; see also Chapter 3 and (3.6). Let

$$c = \bigwedge_{i \in \mathcal{M}} x_i \wedge \bigwedge_{j \in \mathcal{U}} y_j \quad \text{and} \quad h = \bigwedge_{i \in \mathcal{M}} x_i \wedge \bigwedge_{\ell \in \mathcal{R}} w_k \ , \tag{5.1}$$

where $|\mathcal{M}| = m$, $|\mathcal{U}| = u$, and $|\mathcal{R}| = r$. Thus the x's are *mutual* variables, the y's are *undiscovered* variables and the w's are *redundant* variables appearing in h. Variables in the target c are called *good*, and variables not in the target c are called *bad*.

The probability of the error region is

$$\begin{aligned} \varepsilon &= 2^{-m-u} + 2^{-m-r} - 2^{1-m-u-r} \\ &= 2^{-|c|} + 2^{-|h|} - 2^{1-|h|-u} \\ &= 2^{-|c|} + 2^{-|h|}\left(1 - 2^{1-u}\right) . \end{aligned} \tag{5.2}$$

5.3.1 Empty Target

Theorem 5.3.1 (Disagreement Coefficient for Empty Target)**.** *The disagreement coefficient of the empty target is*

$$2 - 2^{1-n} .$$

Proof. When the target is empty, then $m = u = 0$ and (5.2) gives $\varepsilon_\emptyset = 1 - 2^{-r}$, for $r \in \{0, 1, \dots, n\}$. As r increases, the error $\varepsilon_\emptyset$ increases. We will consider these values for error in turn.

First when $\varepsilon_\emptyset = 0$, that is $r = 0$, the only hypothesis that achieves this error is $h = c = \emptyset$. Hence, the disagreement region is empty in this case. However, we are not interested in this particular case, since in the Definition 5.2.3 we only care about positive values of the error.

When $r \geqslant 1$, any truth assignment that is different from the all 1's truth assignment belongs to the disagreement region. To see this consider the first 0 occurrence in that truth assignment and a hypothesis composed of only one variable, specifically that one where we have the first occurrence of 0. Then that particular hypothesis outputs FALSE for this particular truth assignment, while the empty hypothesis outputs TRUE. Then, the number of truth assignments in the disagreement region is $2^n - 1$, which has probability weight equal to $1 - 2^{-n}$. Hence the candidate values for the disagreement coefficient are

$$\frac{1 - 2^{-n}}{1 - 2^{-r}}\,, \text{ for } r \geqslant 1\,.$$

Maximizing the quantity over all the *positive* values of error, that is $r \geqslant 1$, it follows that we want to minimize r. As a consequence we choose $r = 1$. □

5.3.2 Target of Size One

Theorem 5.3.2 (Disagreement Coefficient for Target of Size 1)**.** *The disagreement coefficient of a target of size* 1 *is*

$$2 - 2^{1-n}\,.$$

Proof. Let $c = x$. We distinguish cases based on the number of undiscovered variables. The two possible values are $u = 0$ and $u = 1$.

Error $< 1/2$. In this case the hypotheses have zero undiscovered variables and are extensions of the target with at most $n - 1$ additional variables. Hence, $m = 1, u = 0$ and $r \in \{0, 1, \dots, n - 1\}$, which in turn implies $m + r = |h| \in \{1, 2, \dots, n\}$. Then (5.2) gives the error

$$2^{-1} - 2^{-|h|}\,.$$

Again, the minimum error is 0 when $|h| = 1 \Rightarrow r = 0$ and hence the only hypothesis that achieves this much error is the target. As r increases, so does the error. Moreover, all the truth assignments for which we have a 0 in the position of x do not belong to the disagreement region since all the hypotheses return FALSE as x is falsified. So, we restrict our attention to the truth assignments where $x = 1$. Then, apart from the all 1's truth assignment all the other truth assignments belong to the disagreement region. To see this, consider the first occurrence of 0 in such a truth assignment. Let $y \neq x$ be the variable associated with this 0. Then the hypothesis $h = x \wedge y$ is falsified by this specific truth assignment, while the hypothesis $h = c = x$ returns TRUE. This implies $2^{n-1} - 1$ truth assignments in the disagreement region and hence the first sequence of candidate values for the disagreement coefficient is given by

$$\frac{2^{-1} - 2^{-n}}{2^{-1} - 2^{-|h|}}\,, \quad \text{for } |h| \in \{2, \dots, n\}\,.$$

This ratio is maximized when the denominator is minimized, that is $|h| = 2$, in which case we get $\left(2^{-1} - 2^{-n}\right) / \left(2^{-2}\right) = 2 - 2^{2-n}$.

Error 1/2. The error of the hypotheses of this form is precisely $1/2$ regardless of the number of redundant variables. This value of error is obtained by all the hypotheses that are missing the single variable which belongs to the target. In particular the hypothesis $h = \emptyset$ always returns TRUE. Hence all the 2^{n-1} truth assignments that were not included in the previous case in the disagreement region are now included, since for every one of them the hypothesis $h = \emptyset$ returns TRUE, while the hypothesis

$h = c = x$ returns FALSE for all of them since x is falsified. Again, for the all 1's truth assignment all the hypotheses return TRUE, so this truth assignment is not in the disagreement region again. As a consequence we have $2^n - 1$ truth assignments in the disagreement region and the error is precisely $1/2$. This gives the last candidate for the disagreement coefficient to be

$$\left(1 - 2^{-n}\right) / (1/2) = 2 - 2^{1-n} ,$$

which is larger among all the previous maximum value by an additive factor of 2^{1-n} and hence is the value that we are looking for. □

5.3.3 Target of Maximum Size

The target has size $|c| = n \geqslant 2$. We distinguish cases based on the number of undiscovered variables. Note that a hypothesis of size $|h| = n - \lambda$ always identifies $n - \lambda$ variables from the target. The error of such hypotheses (which are different from the target itself) is given by

$$\text{error}\,(h) = \begin{cases} 2^{-n} & , \quad |h| = n - 1 , \\ 2^{-|h|} - 2^{-n} & , \quad |h| \in \{0, 1, \ldots, n-2\}. \end{cases} \tag{5.3}$$

Lemma 5.3.3 (Disagreement Region for Target of Maximum Size)**.** *For the target of size n and error strictly less than $2^{\lambda-n}$, with $\lambda \in \{1, 2, \ldots, n\}$, the size of the disagreement region is*

$$\sum_{i=1}^{\lambda} \binom{n}{n-i} = -1 + \sum_{i=0}^{\lambda} \binom{n}{n-i} . \tag{5.4}$$

Proof. We study the disagreement region as the error grows. In other words, we split the whole process into n steps, and in each step we allow hypotheses of smaller size to be in the ball of radius ε which determines the disagreement region.

Let us begin with the case where $\lambda = 1$. In this case the error is strictly less than 2^{1-n}. Hence, the hypotheses that we have in a ball of radius ε are all the hypotheses of size $n - 1$. Clearly the truth assignment that is the all 1's truth assignment can not be in the disagreement region because any hypothesis returns TRUE. So for a truth assignment to be in the disagreement region it has to contain at least one 0. Moreover, any truth assignment that has at least two 0's can not belong to the disagreement region for this error because all the hypotheses (plus the target) return FALSE. On the other hand, any truth assignment that has precisely one zero belongs to the disagreement region. To see this note that the target returns FALSE for that particular truth assignment, but there is a hypothesis of size $n - 1$ which satisfies it, namely the hypothesis that is missing the only variable that is 0. Hence, by counting the number of 1's in the truth assignment, the contribution to the disagreement region from this step is

$$\binom{n}{n-1} .$$

The same argument holds for arbitrary $\lambda \in \{1, 2, \ldots, n\}$. So, when the error is *strictly* less than $2^{\lambda-n}$ we are dealing with hypotheses that are of size at least $n - \lambda$. Then any truth assignment that has at least $(n - \lambda)$ 1's can be satisfied by a hypothesis in our ball of radius ε. In particular, the contribution to the disagreement region in every such step of increasing the error is

$$\binom{n}{n-\lambda} .$$

□

Lemma 5.3.4 (Critical Quantity)**.** *Let* $H(x)$ *be the binary entropy of* x*; that is,* $H(x) = -x\lg(x) - (1-x)\lg(1-x)$*. The quantity* $H(x) - x$ *with* $x \in (0, 1/2]$ *is maximized for* $x = 1/3$ *with value* $f_{\max} = H(1/3) - 1/3 \approx 0.5849625$*. Moreover, it holds that* $2^{f_{\max}} = 3/2$*.*

Proof. Let $f(x) = H(x) - x = -x \cdot \lg x - (1-x) \cdot \lg(1-x) - x$, with $x \in (0, 1/2]$. The derivative of f is

$$f'(x) = -\frac{1}{\ln 2} - \lg x + \lg(1-x) + \frac{1}{\ln 2} - 1 = -1 + \lg\left(\frac{1-x}{x}\right) .$$

We observe that for $x = 1/3$ it holds $f'(1/3) = 0$. Moreover,

$$f''(x) = \frac{x}{(1-x)\ln 2} \cdot \frac{(-x-1+x)}{x^2} = -\frac{1}{x(1-x)\ln 2} .$$

In other words, $f''(x) < 0\ \forall x \in (0, 1/2]$. Hence, $f'(x) > 0$ for $x \in (0, 1/3)$, $f'(x) = 0$ for $x = 1/3$, and $f'(x) < 0$ for $x \in (1/3, 1/2]$. As a consequence, the maximum value of f is obtained for $x = 1/3$, which is $f_{\max} = f(1/3) = H(1/3) - 1/3 \approx 0.5849625$.

Finally we note that $2^{f_{\max}} = 2^{-\frac{1}{3}\lg(1/3) - \frac{2}{3}\lg(2/3) - 1/3} = \left(\frac{1}{3}\right)^{-1/3} \cdot \left(\frac{2}{3}\right)^{-2/3} \cdot 2^{-1/3} = \sqrt[3]{3} \cdot \sqrt[3]{\frac{3^2}{2^2}} \cdot \frac{1}{\sqrt[3]{2}} = \frac{\sqrt[3]{3^3}}{\sqrt[3]{2^3}} = \frac{3}{2}$. □

Lemma 5.3.5 (Useful Lower Bounds)**.** *Let* $H(x)$ *be the binary entropy of* x*; that is,* $H(x) = -x\lg(x) - (1-x)\lg(1-x)$ *and let* $n \in \mathbb{N}$ *such that* $n \geqslant 13$*. Moreover, let* $x_0 \in \left(\frac{1}{3}, \frac{1}{3} + \frac{1}{n}\right)$ *and* $x_1 \in \left(\frac{1}{3}, \frac{1}{3} + \frac{1}{n-1}\right)$*. Then it holds*

$$\begin{cases} 2^{(H(x_0) - x_0)n} & > \frac{1}{2} \cdot \left(\frac{3}{2}\right)^n \\ 2^{(H(x_1) - x_1)(n-1)} & > \frac{1}{2} \cdot \left(\frac{3}{2}\right)^{n-1} \end{cases} .$$

Proof. Let $x_0 = 1/3 + y$, with $y \in (0, 1/n)$. Then $2^{(H(x_0) - x_0)n} = 2^{(H(x_0) - 1/3 - y)n} = 2^{(H(x_0) - 1/3)n} \cdot 2^{-yn} > 2^{(H(x_0) - 1/3)n} \cdot 2^{-1}$. Since $n \geqslant 12$, $H(x)$ is monotone increasing for $x \in [1/3, 1/3 + 1/n)$. It follows that the last quantity is at least $\frac{1}{2} \cdot 2^{(H(1/3) - 1/3)n}$ which is $\frac{1}{2} \cdot \left(\frac{3}{2}\right)^n$ due to Lemma 5.3.4.

Let $x_1 = 1/3 + y$, with $y \in (0, 1/(n-1))$. Then $2^{(H(x_1) - x_1)(n-1)} = 2^{(H(x_1) - 1/3 - y)(n-1)} = 2^{(H(x_1) - 1/3)(n-1)} \cdot 2^{-y(n-1)} > 2^{(H(x_1) - 1/3)(n-1)} \cdot 2^{-1}$. Since $n \geqslant 13$, $H(x)$ is monotone increasing for $x \in [1/3, 1/3 + 1/(n-1))$. It follows that the last quantity is at least $\frac{1}{2} \cdot 2^{(H(1/3) - 1/3)(n-1)}$ which is $\frac{1}{2} \cdot \left(\frac{3}{2}\right)^{n-1}$ due to Lemma 5.3.4. □

Theorem 5.3.6 (Disagreement Coefficient for Target of Maximum Size)**.** *For sufficiently large* n*, for the disagreement coefficient* $\rho_{c,\mathcal{U}_n}$ *of the target c of (maximum) size* n *it holds that*

$$\begin{cases} \frac{1}{2(n+1)} \cdot \left(\frac{3}{2}\right)^n < \rho_{c,\mathcal{U}_n} < 2 \cdot \left(\frac{3}{2}\right)^n & , \quad n \pmod 3 = 0 \\ \frac{1}{4(n+1)} \cdot \left(\frac{3}{2}\right)^n < \rho_{c,\mathcal{U}_n} < 2 \cdot \left(\frac{3}{2}\right)^n & , \quad n \pmod 3 \neq 0 \end{cases} .$$

Proof. Similarly to the preceding analysis we consider hypotheses of size $|h| = n - \lambda$.

When $\lambda = 1$ the error is 2^{-n} and the disagreement region has size $\binom{n}{n-1}$. Hence, the first candidate value for the disagreement coefficient is $\frac{n/2^n}{2^{-n}} = n$. However, this value will be dominated.

For larger values of λ, that is $\lambda \in \{2, 3, \ldots, n\}$, we want to maximize the quantity

$$\frac{\left(-1 + \sum_{i=0}^{\lambda} \binom{n}{n-i}\right) \cdot 2^{-n}}{2^{\lambda - n} - 2^{-n}} = \frac{\left(-1 + \sum_{i=0}^{\lambda} \binom{n}{i}\right) \cdot 2^{-n}}{2^{\lambda - n} - 2^{-n}} .$$

First consider values of λ such that $\lambda \geqslant n/2$. The candidate values for the disagreement coefficient based on these values of λ are relatively small compared to what we can achieve with smaller values of λ. In particular, the probability of the disagreement region is never more than 1, and the error is at least $2^{-1-n/2} = \frac{1}{2} \cdot 2^{-n/2} = \frac{1}{2} \cdot \left(\sqrt{2}\right)^{-n}$. It follows that the candidate values in that region can never be more than $\frac{1}{\frac{1}{2}\cdot\left(\sqrt{2}\right)^{-n}} = 2 \cdot \sqrt{2}^{n}$. However, we will see below that the disagreement coefficient, for smaller values of λ, can be $\Omega\left(\frac{1}{n+1} \cdot \left(\frac{3}{2}\right)^{n}\right)$, which is asymptotically larger than $2 \cdot \sqrt{2}^{n}$.

Below we distinguish cases according to the value of $n \pmod 3$.

Upper Bound. For $\lambda = \alpha{\cdot}n$, with $\alpha \in (0, 1/2)$ such that $\alpha{\cdot}n$ is an integer we have $\sum_{i=0}^{\alpha\cdot n}\binom{n}{i} \leqslant 2^{H(\alpha)n}$, and hence by Lemma 5.3.3 the disagreement region has size less than $2^{H(\alpha)n}$. Moreover, for the permissible values of α (that is, $\alpha \cdot n$ is an integer) the error is at least $2^{\alpha\cdot n-n-1} = \frac{1}{2} \cdot 2^{-(1-\alpha)n}$. As a consequence, the candidate values for the disagreement coefficient are less than

$$\frac{2^{H(\alpha)n} \cdot 2^{-n}}{\frac{1}{2} \cdot 2^{-(1-\alpha)n}} = 2 \cdot 2^{(H(\alpha)-1+(1-\alpha))n} = 2 \cdot 2^{(H(\alpha)-\alpha)n} .$$

By Lemma 5.3.4 it follows that no matter what the permissible values of α are, for $\alpha = 1/3$ the disagreement coefficient is for sure less than $2 \cdot \left(\frac{3}{2}\right)^{n}$.

Lower Bound when $n \pmod 3 = 0$. In this case n is a multiple of 3. For $\lambda = \alpha \cdot n$, such that $\lambda \geqslant 2$, the disagreement region has size at least $2^{H(\alpha)n}/(n + 1)$ and moreover the error is less than $2^{1+\alpha\cdot n-n} = 2^{1-(1-\alpha)n}$. As a consequence, the disagreement coefficient is bigger than

$$\frac{1}{n+1} \cdot \frac{2^{H(\alpha)n} \cdot 2^{-n}}{2^{1-(1-\alpha)n}} = \frac{1}{2(n+1)} \cdot 2^{(H(\alpha)-\alpha)n} .$$

By Lemma 5.3.4 it follows that for $\alpha = 1/3$ the disagreement coefficient is bigger than $\frac{1}{2(n+1)} \cdot \left(\frac{3}{2}\right)^{n}$.

Lower Bound when $n \pmod 3 \neq 0$. We select the unique $\alpha_0 \in (1/3, 1/3 + 1/n)$ such that $\lambda = \alpha_0 \cdot n$ is an integer which is at least 2. Then, the disagreement region has size at least $2^{H(\alpha_0)n}/(n + 1)$ and moreover the error is less than $2^{1+\alpha_0\cdot n-n} = 2^{1-(1-\alpha_0)n} < 2^{1-(1-1/3-1/n)n} = 2^{2-2n/3}$. As a consequence, the disagreement coefficient is bigger than

$$\frac{1}{n+1} \cdot \frac{2^{H(\alpha_0)n} \cdot 2^{-n}}{2^{2-2n/3}} = \frac{1}{4(n+1)} \cdot 2^{(H(\alpha_0)-1+2/3)n} > \frac{1}{4(n+1)} \cdot 2^{(H(1/3)-1/3)n} ,$$

where the last inequality follows because $H(x)$ is monotone increasing for $x \in (0, 1/2)$. By Lemma 5.3.4 it follows that the disagreement coefficient is bigger than $\frac{1}{4(n+1)} \cdot \left(\frac{3}{2}\right)^{n}$. □

5.3.4 Target of Size One Less Than Maximum

The target has size $|c| = n - 1$. We study of the disagreement region based on the permissible values of error in increasing order.

Lemma 5.3.7 (Disagreement Region Contribution for Tiny Error (2^{-n}) Hypotheses)**.** *For the target of size $n - 1$ and error at most 2^{-n}, the contribution to the size of the disagreement region is*

$$1 .$$

Proof. Since the error is strictly less than 2^{1-n} which is the weight of the target, we are dealing with hypotheses that are specializations of the target. There is only one such hypothesis, namely the hypothesis $h = c \wedge x$, where x is the only variable missing from the target. As a consequence, the only truth assignment that is introduced into the disagreement region is the truth assignment where x is 0 and all the other variables are equal to 1. □

Lemma 5.3.8 (Disagreement Region Contribution for Small Error $\left(2^{1-n}\right)$ Hypotheses)**.** *For a target of size $n-1$ and error at most 2^{1-n}, the contribution to the size of the disagreement region is*

$$2 \cdot \binom{n-1}{n-2} .$$

Proof. This value of error allows hypotheses that are missing at most one variable from the target. Hence, any truth assignment that has at least two 0's among the good variables does not belong to the disagreement region because all the hypotheses return FALSE. Regarding the truth assignments that have precisely one 0 among the good variables, all of them belong to the disagreement region regardless of whether or not they satisfy the variable x that is missing from the target. Counting the number of 1's in the segment of the truth assignment that deals with good variables gives the lemma. □

Lemma 5.3.9 (Disagreement Region Contribution for Error at most $2^{1+\lambda-n}$ where $\lambda \in \{1, 2, \ldots, n{-}2\}$)**.** *For a target of size $n-1$ and error at most $2^{1+\lambda-n}$, where $\lambda \in \{1, 2, \ldots, n-2\}$, the contribution to the size of the disagreement region at step λ is*

$$\begin{cases} \binom{n-1}{n-3} & , \quad \lambda = 1, \\ \binom{n-1}{n-2-\lambda} + \binom{n-1}{n-1-\lambda} & , \quad \lambda > 1 . \end{cases}$$

Proof. When the error is at most $2^{1-n} + 2^{1+\lambda-n}(1-2^{-\lambda}) = 2^{1+\lambda-n}$ we are dealing with hypotheses of size at least $n-1-\lambda$, where $\lambda \in \{1, 2, \ldots, n-2\}$. Let x be the bad variable.

Let $\lambda = 1$. We have just introduced in our ball the hypotheses of size $n-2$ that are missing $\lambda+1 = 2$ variables from the target. This is only accomplished when the hypotheses have $n-3$ good variables and the variable x. Such hypotheses are satisfied by truth assignments where x as well as the $n-3$ good variables are satisfied. The number of such truth assignments is $\binom{n-1}{n-3}$.

Now fix a $\lambda > 1$. In every such step we are introducing into the disagreement region two kinds of truth assignments. The first kind is similar to the previous case; that is, we have $(n{-}2{-}\lambda)$ 1's among the good variables and moreover x is satisfied. The other kind of truth assignments has $(n{-}1{-}\lambda)$ 1's among the good variables and x is falsified. The number of such truth assignments is $\binom{n-1}{n-2-\lambda} + \binom{n-1}{n-1-\lambda}$. □

Lemma 5.3.10 (Disagreement Region Contribution for Error at Most $1 - 2^{1-n}$)**.** *For a target of size $n-1$ and allowing the maximum error possible $\left(1 - 2^{1-n}\right)$, the contribution to the disagreement region is*

$$1 .$$

Proof. In this last step of the expansion of the error the empty hypothesis also appears in our ball of radius ε. This hypothesis classifies the last truth assignment that has not been introduced into the disagreement region, which is the all 0's truth assignment. (Of course the other truth assignment that has not been introduced into the disagreement region is the all 1's truth assignment, but that particular truth assignment can never be into the disagreement region.) □

Lemma 5.3.11 (Lower Bound for Disagreement Coefficient for Target with Size One Less than Maximum)**.** *Let $n \geqslant 13$. For the disagreement coefficient $\rho_{C,\mathcal{U}_n}$ of the target of size $n-1$ it holds*

$$\begin{cases} \frac{1}{n} \cdot \left(\frac{3}{2}\right)^{n-1} & , \quad n \pmod 3 = 1 \\ \frac{1}{2n} \cdot \left(\frac{3}{2}\right)^{n-1} & , \quad n \pmod 3 \neq 1 \end{cases} .$$

Proof. Let $\lambda \geqslant 2$. We will use Lemmas 5.3.7, 5.3.8, and 5.3.9. When we have error at most $2^{1+\lambda-n}$ the size of the disagreement region is

$$1 + 2 \cdot \binom{n-1}{n-2} + \binom{n-1}{n-3} + \sum_{i=2}^{\lambda} \left(\binom{n-1}{n-2-i} + \binom{n-1}{n-1-i} \right) .$$

Rewriting $1 = 2 \cdot \binom{n-1}{n-1} - 1$ and breaking the second sum we have

$$-1 + 2 \cdot \binom{n-1}{n-1} + 2 \cdot \binom{n-1}{n-2} + \binom{n-1}{n-3} + \sum_{i=2}^{\lambda} \binom{n-1}{n-2-i} + \sum_{i=2}^{\lambda} \binom{n-1}{n-1-i},$$

which is

$$-1 + \binom{n-1}{n-2-\lambda} + 2 \cdot \sum_{i=0}^{\lambda} \binom{n-1}{n-1-i} \geqslant 2 \cdot \sum_{i=0}^{\lambda} \binom{n-1}{n-1-i}.$$

Consider permissible values of α such that $\lambda = \alpha \cdot (n-1)$ is an integer with $\alpha \in (0, 1/2)$ and moreover $\lambda \geqslant 2$. Then the disagreement region is at least

$$2 \cdot \sum_{i=0}^{\lambda} \binom{n-1}{n-1-i} = 2 \cdot \sum_{i=0}^{\lambda} \binom{n-1}{i} > 2 \cdot \binom{n-1}{\lambda} = 2 \cdot \binom{n-1}{\alpha \cdot (n-1)} \geqslant \frac{2}{n} \cdot 2^{H(\alpha)(n-1)}.$$

The error is at most $2^{1+\alpha(n-1)-n} = 2^{\alpha(n-1)-(n-1)} = 2^{-(1-\alpha)(n-1)}$. As a consequence, candidate values for the disagreement coefficient are bigger than

$$\frac{2}{n} \cdot \frac{2^{H(\alpha)(n-1)} \cdot 2^{-n}}{2^{-(1-\alpha)(n-1)}} = \frac{1}{n} \cdot 2^{(H(\alpha)-1+(1-\alpha))(n-1)} = \frac{2^{(H(\alpha)-\alpha)(n-1)}}{n}.$$

Case $n \pmod 3 = 1$. Since $n \pmod 3 = 1$ it follows that $\frac{1}{3} \cdot (n-1) = n/3 - 1/3$ is an integer. Hence, setting $\lambda = 1/3 \cdot (n-1)$, by Lemma 5.3.4 it follows that there are candidate values for the disagreement coefficient that are bigger than $\frac{1}{n} \cdot \left(\frac{3}{2}\right)^{n-1}$.
Case $n \pmod 3 = 2$. Since $n \pmod 3 = 2$ it follows that there is a unique integer in the interval $(n/3, n/3+1)$, namely $(n+1)/3$. We now select α_1 so that λ is equal to that integer; that is, $\alpha_1(n-1) = (n+1)/3 \Rightarrow \alpha_1 = \frac{n+1}{3(n-1)} = \frac{1}{3} + \frac{2/3}{n-1}$. Hence $\alpha_1 \in \left(\frac{1}{3}, \frac{1}{3} + \frac{1}{n-1}\right)$. By Lemma 5.3.5 it follows that $2^{(H(\alpha_1)-\alpha_1)(n-1)} > \frac{1}{2} \cdot \left(\frac{3}{2}\right)^{n-1}$. As a consequence, there are candidate values for the disagreement coefficient which are bigger than $\frac{1}{2n} \cdot \left(\frac{3}{2}\right)^{n-1}$.
Case $n \pmod 3 = 0$. Since $n \pmod 3 = 0$ it follows that $n/3$ is an integer. We now select α_1 so that λ is equal to that integer; that is, $\alpha_1(n-1) = n/3 \Rightarrow \alpha_1 = \frac{n}{3(n-1)} = \frac{1}{3} + \frac{1/3}{n-1}$. Hence $\alpha_1 \in \left(\frac{1}{3}, \frac{1}{3} + \frac{1}{n-1}\right)$. By Lemma 5.3.5 it follows that $2^{(H(\alpha_1)-\alpha_1)(n-1)} > \frac{1}{2} \cdot \left(\frac{3}{2}\right)^{n-1}$. As a consequence, there are candidate values for the disagreement coefficient which are bigger than $\frac{1}{2n} \cdot \left(\frac{3}{2}\right)^{n-1}$. □

Lemma 5.3.12 (Upper Bound for Disagreement Coefficient for Target with Size One Less than Maximum)**.** *The disagreement coefficient of the target of size $n-1$ is at most*

$$4 \cdot \left(\frac{3}{2}\right)^{n-1}.$$

Proof. We examine candidate values for the disagreement coefficient that correspond to increasing values of the error.
Error 2^{-n}. This is the case of Lemma 5.3.7. The probability of the disagreement region is 2^{-n} and the error is 2^{-n}. Hence, the candidate value for the disagreement coefficient is 1.
Error 2^{1-n}. This is the case of Lemma 5.3.8. The size of the disagreement region is $1 + 2 \cdot \binom{n-1}{n-2} = 1 + 2(n-1) = 2n - 1 < 2n$. Hence, the probability of the disagreement region is less than $2n \cdot 2^{-n}$. As a consequence, the candidate value for the disagreement coefficient from this particular value of error is less than $(2n \cdot 2^{-n})/2^{1-n} = n \cdot 2^{1-n-1+n} = n$.

$2^{1-n} <$ **Error** $\leqslant 2^{-1}$**.** This is the case of Lemma 5.3.9. The permissible values of error are separated by considering hypotheses of different sizes, and within such intervals there is an additional refinement on the permissible values for the error which depend on the number of undiscovered variables in the hypotheses of certain size. Since we are aiming for an upper bound, we will overestimate the probability of the disagreement region and we will underestimate the permissible error which allows the disagreement region of interest.

Regarding the big gaps for the error, these are obtained by considering hypotheses of size $n-1-\lambda$, where $\lambda \in \{1, 2, \ldots, n-2\}$.

Now, let $\lambda \geqslant 2$ and $\lambda + 1 = \alpha \cdot (n-1)$, with $\alpha \in (0, 1/2)$. By Lemmas 5.3.7, 5.3.8, and 5.3.9, for a fixed such λ the disagreement region is

$$1 + 2 \cdot \binom{n-1}{n-2} + \binom{n-1}{n-3} + \sum_{i=2}^{\lambda} \left(\binom{n-1}{n-2-i} + \binom{n-1}{n-1-i} \right),$$

which is

$$-1 + \binom{n-1}{n-2-\lambda} + 2 \cdot \sum_{i=0}^{\lambda} \binom{n-1}{n-1-i} < 2 \cdot \sum_{i=0}^{\lambda+1} \binom{n-1}{n-1-i} = 2 \cdot \sum_{i=0}^{\alpha \cdot (n-1)} \binom{n-1}{i}.$$

This last quantity is at most $2 \cdot 2^{H(\alpha)(n-1)}$.

For every particular λ the minimum error is obtained by minimizing the number of undiscovered variables among the hypotheses of the particular size that were just introduced. For any $\lambda \geqslant 2$, the number of undiscovered variables in hypotheses of size $n-1-\lambda$ is at least 2. Hence, by (5.2) the error is at least $2^{1-n} + 2^{-(n-1-\lambda)} \cdot 2^{-1} \geqslant 2^{\lambda - n} = 2^{\alpha \cdot (n-1) - n - 1} = 2^{-2-(1-\alpha)(n-1)}$.

As a consequence of these last observations the candidate values for the disagreement coefficient are less than

$$\frac{2 \cdot 2^{H(\alpha)(n-1)} \cdot 2^{-n}}{2^{-2-(1-\alpha)(n-1)}} = \frac{2^{H(\alpha)(n-1)} \cdot 2^{-(n-1)}}{2^{-2-(1-\alpha)(n-1)}} = 2^2 \cdot 2^{(H(\alpha)-1+(1-\alpha))(n-1)} = 4 \cdot 2^{(H(\alpha)-\alpha)(n-1)}.$$

Hence by Lemma 5.3.4, for $\alpha = 1/3$, the candidate values for the disagreement coefficient are bounded from above by $4 \cdot \left(\frac{3}{2}\right)^{n-1}$.

For larger values of λ, that is $\lambda \geqslant \lceil n/2 \rceil$, the probability of the disagreement region is a constant, and in any case at most 1. On the other hand, the error is at least $2^{-\lfloor n/2 \rfloor - 2}$, and hence the candidate values for the disagreement coefficient are not more than $4 \cdot 2^{\lfloor n/2 \rfloor} \leqslant 4 \cdot \left(\sqrt{2}\right)^n$.

Error $1 - 2^{1-n}$**.** This is the case of Lemma 5.3.10. The probability of the disagreement region is $1 - 2^{-n}$ and the error is $1 - 2^{1-n}$. Hence the disagreement coefficient in this case is $(1 - 2^{-n})/(1 - 2^{1-n}) = 1 + 1/(2^n - 2)$. □

Theorem 5.3.13 (Disagreement Coefficient for Target with Size One Less than Maximum)**.** *For the disagreement coefficient* $\rho_{c,\mathcal{U}_n}$ *of a target of size* $n-1$ *it holds*

$$\begin{cases} \frac{1}{n} \cdot \left(\frac{3}{2}\right)^{n-1} < \rho_{c,\mathcal{U}_n} < 4 \cdot \left(\frac{3}{2}\right)^{n-1} & , \quad n \pmod 3 = 1 \\ \frac{1}{2n} \cdot \left(\frac{3}{2}\right)^{n-1} < \rho_{c,\mathcal{U}_n} < 4 \cdot \left(\frac{3}{2}\right)^{n-1} & , \quad n \pmod 3 \neq 1 \end{cases}.$$

Proof. Immediate from Lemmas 5.3.11 and 5.3.12. □

5.3.5 General Cases

We are now ready to proceed with the general cases.

Theorem 5.3.14 (General Upper Bound on the Disagreement Coefficient)**.** *Consider a target of size* k *such that* $2 \leqslant k \leqslant n-1$*. Then the disagreement coefficient of this target is strictly less than*

$$2^k .$$

Proof. We distinguish cases for the values of error.

Error $< 2^{-k}$**.** First consider values of error strictly smaller than the weight of the target c. By (5.2), such values are possible only when we consider hypotheses that are specializations of the target (plus the target itself); that is $u = 0$ in (5.2). In particular, by (5.2) the minimum non-zero error is obtained by specializations of the target of precisely one more variable and is equal to 2^{-1-k}. On the other hand, the truth assignments that belong to the disagreement region have to satisfy all k variables that appear in the target, otherwise, all the hypotheses return FALSE for these truth assignments. Among the rest $n-k$ positions of the truth assignments, we want to avoid the all 1's extension, since the resulting truth assignment is the all 1's truth assignment and any hypothesis returns TRUE for this truth assignment. On the other hand, if the extension has at least one 0, then the truth assignment belongs to the disagreement region. To see this, consider the first occurrence of 0 in such an extension, and let the variable u be associated with that particular zero. Then the hypothesis $h = c \wedge u$ returns FALSE for this particular truth assignment while c classifies this particular truth assignment as TRUE.

Moreover, as the error ε increases, but is still less than 2^{-k}, all that really happens is that we allow more hypotheses in our balls of radius ε, but the disagreement region remains unchanged. Hence, the candidate values for the disagreement coefficient drop as ε increases.

In any case, the number of truth assignments in the disagreement region is $2^{n-k}-1$ and the error is at least 2^{-1-k}. Hence, the maximum candidate value for the disagreement coefficient among errors that are strictly less than 2^{-k} is

$$\frac{\left(2^{n-k}-1\right)\cdot 2^{-n}}{2^{-1-k}} = \frac{2^{-k}-2^{-n}}{2^{-1-k}} = 2 - 2^{k+1-n} .$$

Error 2^{-k}**.** Let the error be precisely 2^{-k}. Then, the hypotheses in our ball of radius ε are all the hypotheses that we have in the previous case plus the hypotheses that are missing precisely one variable from the target. First we note that any truth assignment that has less than $(k-1)$ 1's among the k variables that appear in the target is classified as FALSE by any hypothesis in our ball, and hence all these truth assignments do not belong to the disagreement region. On the other hand, any truth assignment that has precisely one 0 among these k positions belongs to the disagreement region. To see this note that the target classifies all such truth assignments as FALSE, while on the other hand we have at our disposal a hypothesis that can classify this particular truth assignment as TRUE. Hence, the total number of truth assignments that belong to the disagreement region is $\binom{k}{k}\cdot 2^{n-k}-1+\binom{k}{k-1}\cdot 2^{n-k} = (k+1)\cdot 2^{n-k}-1$. As a consequence, the candidate value for the disagreement coefficient in this case is

$$\frac{\left((k+1)\cdot 2^{n-k}-1\right)\cdot 2^{-n}}{2^{-k}} = (k+1) - 2^{k-n} .$$

Error $> 2^{-k}$**.** Let the error be strictly more than 2^{-k}. Since, the probability of the disagreement region is at most 1, candidate values for the disagreement coefficient for these values of error are strictly less than

$$\frac{1}{2^{-k}} = 2^k .$$

The claim follows from this last case. □

Lemma 5.3.15 (Lower Bound on the Disagreement Coefficient of Short Targets)**.** *Consider a target of size* k *such that* $2 \leqslant k \leqslant \lfloor n/2 \rfloor$*. Then, the disagreement coefficient of this target is more than*

$$\frac{1}{4} \cdot 2^k \,.$$

Proof. For a target c of size k such that $2 \leqslant k \leqslant \lfloor n/2 \rfloor$, let $\mathcal{H}^k_{\geqslant 2}$ be the set of hypotheses that are missing missing at least least two variables from the target and are of size at least k. Moreover, let $\mathcal{H}_1$ be the set of hypotheses that are missing precisely one variable from the target, and note that all these hypotheses have size at least $k-1$. Finally, let $\mathcal{H}_0$ be the set of hypotheses that are missing no variables from the target. Note that all these hypotheses in $\mathcal{H}_0$ are of size at least k. Let the maximum error obtained from the hypotheses in $\mathcal{H}^k_{\geqslant 2}$ be ε. These hypotheses form our ball of radius $\mathbf{B}_{\mathcal{U}}(c, \varepsilon) = \mathcal{H}_0 \cup \mathcal{H}_1 \cup \mathcal{H}^k_{\geqslant 2}$. We now study the disagreement region for this particular error ε.

- Pick a random truth assignment σ that has at least k 1's. Then, as long as

  $$\sigma \neq \underbrace{11\ldots11}_{\text{all 1's}},$$

  the claim is that this truth assignment belongs to the disagreement region.

  - If $c(\sigma)$ is FALSE, then it is easy to find a hypothesis that satisfies this truth assignment. The reason is that σ has at least k 1's and we have at our disposal all the hypotheses of size at least k. In particular we take the hypothesis that contains the variables that are set to 1 in the truth assignment.
  - If $c(\sigma)$ is TRUE, recall that σ has at least one 0. Let the variable of the first occurrence of 0 in σ be the variable x. Then the hypothesis $h = c \wedge x$ classifies σ as FALSE.

  The number of truth assignments considered in this case are $-1 + \sum_{i=k}^{n} \binom{n}{i}$.

- Now pick a random truth assignment σ that has at most $(k-2)$ 1's. The claim is that σ does not belong to the disagreement region. Clearly $c(\sigma)$ is FALSE. Moreover, all the hypotheses in our ball of radius ε contain hypotheses of size at least $k-1$. Hence, any such hypothesis h also returns FALSE as at least one variable is falsified in every one of them. The number of truth assignments considered in this case are $\sum_{i=0}^{k-2} \binom{n}{i}$.

- Finally pick a random truth assignment σ that has precisely $(k-1)$ 1's. Clearly $c(\sigma)$ is FALSE. In order for such a truth assignment to belong to the disagreement region we must be able to find a hypothesis such that $h(\sigma)$ is TRUE. The only hypotheses that we have of size $k-1$ are the hypotheses of that particular size that are missing precisely one variable from the target c. Hence, among the $\binom{n}{k-1}$ truth assignments considered in this case, only $\binom{k}{k-1} = k$ belong to the disagreement region.

By the above analysis, the number of truth assignments that belong to the disagreement region is

$$k - 1 + \sum_{i=k}^{n} \binom{n}{i} \geqslant \sum_{i=k}^{n} \binom{n}{i} \geqslant \sum_{i=\lfloor n/2 \rfloor}^{n} \binom{n}{i} \geqslant 2^{n-1} \,.$$

In other words, the probability of the disagreement region is at least $1/2$.

On the other hand, by (5.2) the error for the above analysis is strictly less than $2^{-k} + 2^{-k} = 2^{1-k}$. As a consequence, the disagreement coefficient for targets of size k such that $2 \leqslant k \leqslant \lfloor n/2 \rfloor$, is

$$\rho > \frac{1/2}{2^{1-k}} = \frac{1}{4} \cdot 2^k \,,$$

which implies the statement of the lemma. □

Theorem 5.3.16 (Disagreement Coefficient for Short Targets)**.** *Consider a target* c *of size* k *such that* $2 \leqslant k \leqslant \lfloor n/2 \rfloor$. *Then, for the disagreement coefficient* $\rho_{c,\mathcal{U}_n}$ *of this target it holds*

$$\frac{1}{4} \cdot 2^k < \rho_{c,\mathcal{U}_n} < 2^k\,.$$

Proof. Immediate from Theorem 5.3.14 and Lemma 5.3.15. □

Lemma 5.3.17 (Lower Bound on the Disagreement Coefficient of Long Targets)**.** *Consider a target* c *of size* k *such that* $\lfloor n/2 \rfloor + 1 \leqslant k \leqslant n-2$. *Then, for the disagreement coefficient* $\rho_{c,\mathcal{U}_n}$ *of this target it holds*

$$\rho_{c,\mathcal{U}_n} > \begin{cases} \frac{1}{2(n+1)} \cdot 2^{(H(1-k/n)-(1-k/n))n} & , \quad \lfloor n/2 \rfloor < k \leqslant 2n/3 \\ \frac{1}{4(n+1)} \cdot \left(\frac{3}{2}\right)^n & , \quad 2n/3 < k \leqslant n-2 \quad \textit{and} \quad n \pmod 3 = 0 \\ \frac{1}{8(n+1)} \cdot \left(\frac{3}{2}\right)^n & , \quad 2n/3 < k \leqslant n-2 \quad \textit{and} \quad n \pmod 3 \neq 0 \end{cases}.$$

Proof. Let $k = n - \alpha n$, with $\alpha \in (0, 1/2)$. For reasons that will soon be apparent we distinguish cases when $\lfloor n/2 \rfloor + 1 \leqslant k \leqslant 2n/3$ and $2n/3 < k \leqslant n-2$.

Size k **such that** $\lfloor n/2 \rfloor < k \leqslant 2n/3$**.** Note that in this case $\alpha \in [1/3, 1/2)$. We consider the value of error that is obtained by hypotheses of size k. Note that for this value of error we have hypotheses in our ball of radius ε that are missing at least two variables. Moreover for this value of error we also have at our disposal the hypotheses that are missing precisely 0 or 1 variable from the target. In particular, these last hypotheses have size at least k and $k-1$ respectively.

The proof is similar to that of Lemma 5.3.15.

- A truth assignment that has at least k 1's and is different from the all 1's truth assignment belongs to the disagreement region.
- A truth assignment that has at most $(k-2)$ 1's does not belong to the disagreement region.
- Finally, among the $\binom{n}{k-1}$ truth assignments that have precisely $(k-1)$ 1's, only $\binom{k}{k-1} = k$ belong to the disagreement region.

Hence, the number of truth assignments that belong to the disagreement region is

$$k - 1 + \sum_{i=k}^{n} \binom{n}{i} \geqslant \sum_{i=k}^{n} \binom{n}{i} = \sum_{(1-\alpha)n}^{n} \binom{n}{i} = \sum_{i=0}^{\alpha \cdot n} \binom{n}{i} \geqslant \frac{2^{H(\alpha)n}}{n+1}\,.$$

By (5.2) the error is strictly less than $2 \cdot 2^{-k} = 2^{1-(1-\alpha)n}$. As a consequence, for the disagreement coefficient it holds

$$\begin{aligned} \rho_{c,\mathcal{U}_n} &> \frac{\frac{2^{H(\alpha)n}}{n+1} \cdot 2^{-n}}{2^{1-(1-\alpha)n}} \\ &= \frac{1}{2(n+1)} \cdot 2^{(H(\alpha)-1+(1-\alpha))n} \\ &= \frac{1}{2(n+1)} \cdot 2^{(H(\alpha)-\alpha)n} \end{aligned}$$

Size k **such that** $2n/3 < k \leqslant n-2$**.** Note that $\alpha \in (0, 1/3)$. We consider the value of error that is obtained by hypotheses of size $s = k - \beta n - 1 = n - (\alpha+\beta)n - 1 \leqslant k-1$, such that $\alpha + \beta < 1/2$. Note that for this value of error we have hypotheses in our ball of radius ε that are missing at least two variables. Moreover for this value of error we also have at our disposal the hypotheses that are missing precisely 0 or 1 variable from the target. In particular, these last hypotheses have size at least k and $k-1$ respectively.

Again, the proof is similar to that of Lemma 5.3.15.

- A truth assignment that has at least s 1's and is different from the all 1's truth assignment belongs to the disagreement region.
- A truth assignment that has at most $(s-1)$ 1's does not belong to the disagreement region.

Hence, the number of truth assignments that belong to the disagreement region is

$$-1+\sum_{i=s}^{n}\binom{n}{i} = -1+\binom{n}{s}+\sum_{i=s+1}^{n}\binom{n}{i} \geqslant \sum_{i=n-(\alpha+\beta)n}^{n}\binom{n}{i} = \sum_{i=0}^{(\alpha+\beta)n}\binom{n}{i} \geqslant \frac{2^{H(\alpha+\beta)n}}{n+1}.$$

By (5.2) the error is dominated by the weight of the minimum size hypotheses. The error is strictly less than $2\cdot 2^{-s}$. Since s is an integer and moreover $s=(1-(\alpha+\beta))n-1$, which is strictly less than k, by (5.2) it follows that the error is less than $2\cdot 2^{1-(1-(\alpha+\beta))n} = 2^{2-(1-(\alpha+\beta))n}$. Recall that $k=n-\alpha n \Rightarrow \alpha=(n-k)/n$. Hence, $s=n-(n-k)-\beta n-1=k-\beta n-1$, where β is such so that $\beta\cdot n$ is an integer. As a consequence, for the disagreement coefficient it holds

$$\begin{aligned}\rho_{c,\mathcal{U}_n} &> \frac{\frac{2^{H(\alpha+\beta)n}}{n+1}\cdot 2^{-n}}{2^{2-(1-(\alpha+\beta))n}}\\ &= \frac{1}{4(n+1)}\cdot 2^{(H(\alpha+\beta)-1+(1-(\alpha+\beta)))n}\\ &= \frac{1}{4(n+1)}\cdot 2^{(H(\alpha+\beta)-(\alpha+\beta))n}\end{aligned}$$

Before we continue with the proof we observe that $\alpha\cdot n$ is an integer by the definition of α.

Case $n \pmod 3 = 0$. We set $\beta=1/3-\alpha=1/3-(1-k/n)=k/n-2/3$. Then, $\beta\cdot n$ is also an integer since $n \pmod 3=0$. Moreover, $\alpha+\beta=1/3$. Hence, by Lemma 5.3.4 there are candidate values for the disagreement coefficient for this case that are bigger than $\frac{1}{4(n+1)}\cdot\left(\frac{3}{2}\right)^n$.

Case $n \pmod 3 = 1$. We set $\beta=\frac{k}{n}-\frac{2}{3}+\frac{2}{3n}$. First note that $\beta\cdot n=k-2n/3+2/3=k-2\cdot\frac{(n-1)}{3}$ which is an integer since k is an integer and $(n-1) \pmod 3=0$. Moreover, $\alpha+\beta=1-k/n+k/n-2/3+2/(3n)=\frac{1}{3}+\frac{2/3}{n}$. By Lemma 5.3.5 it follows that there are candidate values for the disagreement coefficient which are bigger than $\frac{1}{8(n+1)}\cdot\left(\frac{3}{2}\right)^n$.

Case $n \pmod 3 = 2$. We set $\beta=\frac{k}{n}-\frac{2}{3}+\frac{1}{3n}$. First note that $\beta\cdot n=k-2n/3+1/3=k-\frac{2n-1}{3}$ which is an integer since k is an integer and $2n-1 \pmod 3=0$. Moreover, $\alpha+\beta=1-k/n+k/n-2/3+1/(3n)=\frac{1}{3}+\frac{1/3}{n}$. By Lemma 5.3.5 it follows that there are candidate values for the disagreement coefficient which are bigger than $\frac{1}{8(n+1)}\cdot\left(\frac{3}{2}\right)^n$. □

Lemma 5.3.18 (Upper Bound on the Disagreement Coefficient of Long Targets)**.** *Consider a target c of size k such that $\lfloor n/2\rfloor+2\leqslant k\leqslant n-2$. Then, for the disagreement coefficient $\rho_{c,\mathcal{U}_n}$ of this target it holds*

$$\rho_{c,\mathcal{U}_n} < \begin{cases} 2\cdot 2^{(H(1-k/n)-(1-k/n))n}, & \lfloor n/2\rfloor+2 \leqslant k \leqslant 2n/3\\ 2\cdot\left(\frac{3}{2}\right)^n, & 2n/3 < k \leqslant n-2\end{cases}.$$

Proof. We examine the candidate values for the disagreement coefficient for increasing values of the error.

Error $< 2^{-k}$. The maximum candidate value for the disagreement coefficient among errors that are strictly less than 2^{-k} is $2-2^{k+1-n}$. See the equivalent case in Theorem 5.3.14.

Error 2^{-k}. The candidate value for the disagreement coefficient in this case is $(k+1)-2^{k-n}$. Again see the equivalent case in Theorem 5.3.14.

Error $> 2^{-k}$. This is the important case from where we will obtain the upper bound. Let $k=n-\alpha n$, with $\alpha\in(0,1/2)$. We distinguish cases when $\lfloor n/2\rfloor+2\leqslant k\leqslant 2n/3$ and $2n/3<k\leqslant n-2$ similarly to Lemma 5.3.17.

Size k **such that** $\lfloor n/2 \rfloor + 2 \leqslant k \leqslant 2n/3$**.** Note that in this case $\alpha \in [1/3, 1/2)$. As long as we consider increasing values of error that are less than $2 \cdot 2^{-k} = 2^{1-k}$, we are introducing in our ball of radius ε hypotheses that are missing at least two variables from the target with successively smaller sizes but their sizes is at least k. During this process, the disagreement region increases, but in any case is bounded from above by the size of the disagreement region that is formed when we introduce hypotheses of size at least k which are missing as many variables from the target as possible. In that case the disagreement region is (see for example Lemma 5.3.17)

$$k - 1 + \sum_{i=k}^{n} \binom{n}{i} < n + \sum_{i=(1-\alpha)n}^{n} \binom{n}{i} = n + \sum_{i=0}^{\alpha n} \binom{n}{i} \leqslant n + 2^{H(\alpha)n} < 2^{1+H(\alpha)n},$$

where the last inequality follows since $H(\alpha) > 9/10$ for $\alpha \in [1/3, 1/2)$ and $\lg n < 9n/10$ for every $n \geqslant 2$. As a consequence, the candidate values for the disagreement coefficient in this region of error are less than

$$\frac{2^{1+H(\alpha)n} \cdot 2^{-n}}{2^{-k}} = \frac{2^{1+H(\alpha)n} \cdot 2^{-n}}{2^{-(1-\alpha)n}} = 2 \cdot 2^{(H(\alpha)-1+(1-\alpha))n} = 2 \cdot 2^{(H(\alpha)-\alpha)n}.$$

Now let us consider values of error that are at least 2^{1-k} and less than $2^{-k} + 2^{-(\lfloor n/2 \rfloor + 1)}$. By (5.2) any such permissible value of error belongs to an interval of the form $[2^{-k} + 2^{-1+\lambda-k}, 2^{-k} + 2^{\lambda-k})$, where $k - \lambda$ is the minimum size of the hypotheses. Moreover, let $\lambda = \beta n$ such that $(\alpha + \beta)n \leqslant \lfloor n/2 \rfloor$. The disagreement region for such an error is bounded from above by the disagreement region of the maximum error that is less than $2^{-k} + 2^{\lambda-k}$. Excluding the all 1's truth assignment, the disagreement region is thus bounded from above by the number of truth assignments that have at least $(k - \lambda)$ 1's which is $-1 + \sum_{i=k-\lambda}^{n} \binom{n}{i} \leqslant \sum_{i=0}^{n-k+\lambda} \binom{n}{i} = \sum_{i=0}^{\alpha n + \beta n} \binom{n}{i} \leqslant 2^{H(\alpha+\beta)n}$. For every such λ the error is at least $2^{-k} + 2^{-1+\lambda-k} > 2^{-1+\lambda-k} = 2^{-1+\beta n-(1-\alpha)n} = 2^{-1-(1-(\alpha+\beta))n}$. It follows that the candidate values for the disagreement coefficient in this region of error are less than

$$\frac{2^{H(\alpha+\beta)n} \cdot 2^{-n}}{2^{-1-(1-(\alpha+\beta))n}} = 2 \cdot 2^{(H(\alpha+\beta)-1+(1-(\alpha+\beta)))n} = 2 \cdot 2^{(H(\alpha+\beta)-(\alpha+\beta))n}.$$

However, by Lemma 5.3.4 the function $H(x) - x$ is monotone decreasing for $x \in (1/3, 1/2]$. Since $\alpha \geqslant 1/3$ and $\alpha + \beta \leqslant 1/2$, it follows that β should be minimized, and hence the candidate values for the disagreement coefficient are less than $2 \cdot 2^{(H(\alpha)-\alpha)n}$.

Finally consider values of error that are at least $2^{-k} + 2^{-1-(\lfloor n/2 \rfloor + 1)}$. Then the probability of the disagreement region is at most 1 and the error is at least $2^{-2-\lfloor n/2 \rfloor}$. As a consequence the candidate values for the disagreement coefficient are less than $4 \cdot 2^{\lfloor n/2 \rfloor}$. On the other hand, the function $H(x) - x$ is monotone decreasing for $x \in [1/3, 1/2]$, with minimum value $H(1/2) - 1/2 = 1 - 1/2 = 1/2$. Therefore, since $\alpha < 1/2 \Rightarrow H(\alpha) - \alpha > 1/2$, for sufficiently large n it holds that $4 \cdot 2^{\lfloor n/2 \rfloor} \leqslant 2 \cdot 2^{(H(\alpha)-\alpha)n}$.

Size k **such that** $2n/3 < k \leqslant n - 2$**.** Note that in this case $\alpha \in (0, 1/3)$. Again, in the first case we consider increasing values of error that are less than $2 \cdot 2^{-k} = 2^{1-k}$. The size of the disagreement region is again bounded from above by the quantity

$$k - 1 + \sum_{i=k}^{n} \binom{n}{i}.$$

However, this time $\alpha \in (0, 1/3)$ and hence we can not give the same bound as before. On the other hand, it holds

$$k - 1 + \sum_{i=k}^{n} \binom{n}{i} < \binom{n}{k-1} + \sum_{i=k}^{n} \binom{n}{i} = \sum_{i=(1-\alpha)n-1}^{n} \binom{n}{i} = \sum_{i=0}^{\alpha n+1} \binom{n}{i} \leqslant 2^{H(\alpha+1/n)n}.$$

Regarding the error, it is at least $2^{-k} = 2^{-(1-\alpha)n}$ and hence the candidate values for the disagreement coefficient are less than

$$\frac{2^{H(\alpha+1/n)n} \cdot 2^{-n}}{2^{-(1-\alpha)n}} = 2^{(H(\alpha+1/n)-1+(1-\alpha))n} = 2 \cdot 2^{(H(\alpha+1/n)-(\alpha+1/n))n} .$$

Now let us consider values of error that are at least 2^{1-k} and less than $2^{-k} + 2^{-(\lfloor n/2 \rfloor+1)}$. Again by (5.2) any such permissible value of error belongs to an interval of the form $[2^{-k} + 2^{-1+\lambda-k}, 2^{-k} + 2^{\lambda-k})$, where $k - \lambda$ is the minimum size of the hypotheses. Moreover, let $\lambda = \beta n$ such that $(\alpha + \beta)n \leqslant \lfloor n/2 \rfloor$. The disagreement region for such an error is bounded from above by the disagreement region of the maximum error that is less than $2^{-k} + 2^{\lambda-k}$. Excluding the all 1's truth assignment, the disagreement region is thus bounded from above by the number of truth assignments that have at least $(k - \lambda)$ 1's which is $-1 + \sum_{i=k-\lambda}^{n} \binom{n}{i} \leqslant \sum_{i=0}^{n-k+\lambda} \binom{n}{i} = \sum_{i=0}^{\alpha n+\beta n} \binom{n}{i} \leqslant 2^{H(\alpha+\beta)n}$. For every such λ the error is at least $2^{-k} + 2^{-1+\lambda-k} > 2^{-1+\lambda-k} = 2^{-1+\beta n-(1-\alpha)n} = 2^{-1-(1-(\alpha+\beta))n}$. It follows that the candidate values for the disagreement coefficient in this region of error are less than

$$\frac{2^{H(\alpha+\beta)n} \cdot 2^{-n}}{2^{-1-(1-(\alpha+\beta))n}} = 2 \cdot 2^{(H(\alpha+\beta)-1+(1-(\alpha+\beta)))n} = 2 \cdot 2^{(H(\alpha+\beta)-(\alpha+\beta))n} .$$

However, by Lemma 5.3.4 the function $H(x) - x$ achieves its maximum when $\alpha + \beta = 1/3$. As a consequence, the candidate values for the disagreement coefficient are bounded from above by the quantity

$$2 \cdot 2^{(H(1/3)-1/3)n} = 2 \cdot \left(\frac{3}{2}\right)^n .$$

Finally, as the error increases even further, the candidate values that we obtain for an upper bound are smaller. Similarly to the case before, we obtain an upper bound of the form $\mathcal{O}\left(2^{\lfloor n/2 \rfloor}\right)$, which is asymptotically smaller than the $\mathcal{O}\left(\left(\frac{3}{2}\right)^n\right)$ that we have from the previous interval of errors.

Hence, in the case where $\lfloor n/2 \rfloor + 2 \leqslant k \leqslant 2n/3$ the upper bound is $2 \cdot 2^{(H(1-k/n)-(1-k/n))n}$, while when $2n/3 < k \leqslant n - 2$ the upper bound is $2 \cdot \left(\frac{3}{2}\right)^n$. Note that if n is divisible by 3 then for $k = 2n/3$ the two bounds are the same. □

Theorem 5.3.19 (Disagreement Coefficient for Long Targets)**.** *Consider a target* c *of size* k *such that* $\lfloor n/2 \rfloor + 1 \leqslant k \leqslant n - 2$. *Let* $\alpha = 1 - k/n$. *Then, for the disagreement coefficient* $\rho_{c,\mathcal{U}_n}$ *of this target it holds*

$$\begin{cases} \frac{2^{(H(\alpha)-\alpha)n}}{2\cdot(n+1)} & < & \rho_{c,\mathcal{U}_n} & < & 2^k & , & k = \lfloor n/2 \rfloor + 1 \\ \frac{2^{(H(\alpha)-\alpha)n}}{2\cdot(n+1)} & < & \rho_{c,\mathcal{U}_n} & < & 2^{1+(H(\alpha)-\alpha)n} & , & \lfloor n/2 \rfloor + 2 \leqslant k \leqslant 2n/3 \\ \frac{(3/2)^n}{4\cdot(n+1)} & < & \rho_{c,\mathcal{U}_n} & < & 2 \cdot (3/2)^n & , & 2n/3 < k \leqslant n-2 \quad \textit{and} \quad \upsilon = 0 \\ \frac{(3/2)^n}{8\cdot(n+1)} & < & \rho_{c,\mathcal{U}_n} & < & 2 \cdot (3/2)^n & , & 2n/3 < k \leqslant n-2 \quad \textit{and} \quad \upsilon \neq 0 \end{cases} ,$$

where $\upsilon = n \pmod 3$.

Proof. For the lower bounds we use Lemma 5.3.17 in every case. When $k = \lfloor n/2 \rfloor + 1$ we use Theorem 5.3.14 for the upper bound, while, for all the other values of k we use Lemma 5.3.18 for the upper bound. □

Chapter 6

Network Analysis of Knowledge Bases

Enabling computers to do common-sense reasoning is one of the basic challenges in AI, implicit in the work of Turing, and made explicit by McCarthy [93]. The availability of large amounts of common-sense knowledge is widely accepted to be a necessary condition for such reasoning. It is an important and relatively recent development that large common-sense knowledge bases such as Cyc and ConceptNet are publicly available. This availability, combined with potential new applications in web search, robotics, human-computer interaction and other areas, has led to an increased interest in common-sense reasoning.

ConceptNet, on which we focus in this chapter, is a semantic net of triples of the form (`concept`$_1$, `relation`, `concept`$_2$), with every `relation` coming from a fixed set of about two dozen relations, such as `IsA` and `Causes` [67, 66]. Its data was initially collected via the web in the form of sentences, and turned into statements using NLP tools. The triples are represented as a sparse matrix with concepts as rows and relation-concept pairs as columns. Low-rank approximations of a condensed version of the matrix, called AnalogySpace, are also available [127]. Thus, ConceptNet allows for *both symbolic and statistical reasoning.*

ConceptNet has been used for various applications (e.g., [123]), including query answering [79]. Moreover, ConceptNet was recently evaluated for its ability in *answering IQ-tests for children*, which was proposed as a general evaluation method for common-sense knowledge bases. ConceptNet's verbal IQ corresponded to an average 4-year old [101, 102]. The algorithms used for answering the IQ-test items were quite simple; that work was also intended to evaluate the ease of use of the system.

Thus, the ConceptNet system is an example of a large knowledge base that has had at least some success in some common-sense tasks, and is a suitable target for studying properties of large common-sense knowledge bases.

6.1 Network Analysis of Knowledge Bases

Here we explore the properties of ConceptNet using the tools of *network analysis* [18, 44, 99]. Social, collaboration, and information networks are major well-studied classes of networks. Network analysis has been applied also to biological, infrastructure, and transportation networks. *Knowledge networks*, such as word association networks and WordNet have also been studied to some extent [33, 130]. The recent work of [64] on automated generation is a potential source of many new networks.

To date, however, there appears to be no general understanding of the characteristics of large knowledge networks. The availability of *ground truth* in common-sense knowledge networks, provided by the meaning of concepts, is an unusual, useful feature for network analysis. Ground truth, for example, allows one to evaluate and compare the quality of groupings found by various community inference algorithms. For example, the reader can immediately see that the community of concepts in

Figure 6.4 (inferred by the clique percolation algorithm [34]) is meaningful, while in a social network it is typically difficult to tell whether a group of people is, in fact, a community.

A detailed network analysis of ConceptNet is given in [38]. Here we give a small sample of the results. There are many different ConceptNet-based networks to consider (directed/undirected edges, multiple edges and loops allowed or not, all relations are considered or just a specific subset of them). In general, those networks have a highly skewed degree distribution and the small world property [1]. Cores form a nested structure of increasing density, similar to other large networks [85]. It is likely that an inner core contains more important concepts, and those could perhaps be given closer attention for additional processing. Among the many community finding algorithms [109, 100, 106, 53, 22, 143, 13, 108, 112] that are implemented in igraph [28] as well as the clique percolation community finding algorithm of Palla et al. [34] which is implemented as `CFinder` [104], clique percolation for appropriate clique sizes seems to give interesting communities. These communities can be useful for finding missing entries and identifying new concepts. The seminal paper [130] proposed cognitive science applications of network information for semantic networks, such as relevance for the age of acquisition of concepts. Cores, communities and other specific structures found by network analysis could be of interest in this context as well. There is theoretical computer science work on exploiting structural properties to get faster algorithms for problems which are hard for large networks (see, e.g., [56]). Such problems include versions of centrality which may also be relevant for cognitive science [33].

6.2 Potential Benefits for Common-Sense Reasoning

We believe the network analysis of ConceptNet and other such networks has potential benefits for common-sense reasoning applications of ConceptNet. The most difficult question type for ConceptNet in the IQ testing was Comprehension, which tests the comprehension of concepts using *why*-questions like Why do we put on sunscreen in summer? Answering WPPSI-III Comprehension questions has overlap with the area of open-domain question answering, of *Jeopardy!* fame, which involves information retrieval, natural language processing and human-computer interaction [91]. In general, however, knowledge representation and reasoning are often weak spots for question answering [11, p. 780], and those abilities seem to be absolutely necessary to answer questions like the one about sunscreen. Incidentally, answering specifically *why*-questions is considered a difficult task [142].

Answering *why*-questions with ConceptNet remains an important and interesting challenge and it serves as one motivation for the explorations described in this chapter. Improving the results and being able to answer test questions for older children is likely to require using *more involved test-answering algorithms* and *improving and enhancing the knowledge base*, for example, by adding missing entries, correcting incorrect entries and providing additional knowledge. Additional knowledge could include additional facts, but also new general knowledge, and capabilities for doing different forms of common-sense reasoning. These issues are also discussed in the papers on ConceptNet [67, 66, 127]; here we propose some further approaches.

ConceptNet provides spreading activation as a tool to find semantically related concepts. This, in turn, can be used as a tool for question answering. The answers obtained using spreading activation are often meaningful, especially if one considers not only the highest ranked answer, but also the best answer among the highest ranked ones. This suggests refined search procedures, where one analyzes the detailed results of spreading activation to rank candidate answers and to identify errors.

Moreover, ConceptNet provides a rich body of knowledge about similarity, ontologies, causality and other notions. It would be useful to enhance this body of knowledge adding further reasoning tools. As a first step, one could build a 'microtheory' of the relations used, for example, that IsA is transitive: (`a`, `IsA`, `b`), (`b`, `IsA`, `c`) → (`a`, `IsA`, `c`). As there are a large number of possible rules, one could try to mine

[1] A graph has the *small-world property* if the distance between two randomly chosen nodes is small, typically logarithmic in the number of nodes.

ConceptNet for such rules. We mention some results of rule mining and give observations for possible applications.

6.3 ConceptNet 4

By "ConceptNet" [67, 66, 127] we mean specifically the version of ConceptNet 4 released in March 2012. In fact, there are two versions of ConceptNet. One, which we call the *large graph*, contains roughly 280,000 English-language concepts, and is released in SQLite database format. The other, which we call the *small graph*, contains roughly 22,000 concepts, and is released as part of a Python package called Divisi, which in general is "a general-purpose tool for reasoning over semantic networks"(`http://csc.media.mit.edu/analogyspace`) and working with large sparse matrices [126]. When unspecified, in this paper we refer to the large graph. Divisi also contains tools for creating truncated singular value decomposition (SVD) forms of the small graph, which its authors refer to as AnalogySpace. The work on IQ-testing ConceptNet [101, 102] was done primarily with AnalogySpace and made no use of the large graph.

Both the large graph and the small graph are sparse, with the large graph being considerably sparser. The small graph was formed from the large graph by dropping some combination of triples that had relatively few users supporting them and concepts that had very little connectivity to the rest of the graph. (The AnalogySpace graph, which we do not discuss here, is dense.)

ConceptNet triples are called *assertions*. Each assertion also has a *score, frequency*, and *polarity*. The score measures the reliability of an assertion, based on the amount of user support it received. Frequency expresses how often the assertion is true, in the range of "never" to "always". Polarity is a coarse-grained version of the frequency and is positive or negative. For example the statement Penguins are not capable of flying has negative polarity. Roughly 3.5% of assertions have negative polarity. Associated with each assertion there is at most one *sentence* and *raw assertion*. The sentence is actual user input that generated or supported the assertion, and the raw assertion is a lightly processed sentence put into one of a large number of standard frames.

6.4 Network Analysis of ConceptNet 4

The prevalence of large networks such as the Web, the internet and online social networks, has led to the explosive growth of research and the development of computational approaches for network analysis and algorithms for large networks, with central concepts such as highly skewed node degree distribution, small world property [144] and algorithms like PageRank [103]. The main insight gained is that, perhaps surprisingly, networks coming from completely different disciplines have quite similar structural properties. In this section we apply this methodology to ConceptNet with an eye toward exploiting its properties for knowledge base algorithms. We expect similar properties to hold for future versions of ConceptNet and other knowledge bases as well. We use `igraph` [28] for most of the network analysis tasks, `CFinder` [104] for computing communities by percolating cliques, and the software that is available online (`http://tuvalu.santafe.edu/~aaronc/powerlaws/`) for the *maximum likelihood estimate (MLE)* for power law fitting described in [23].

Among the many possibilities for viewing ConceptNet as a network, for *degree distribution* we consider the directed multigraph with self-loops formed by assertions with a positive score (and arbitrary polarity). There are $279,497$ concepts appearing in such assertions in the English language version. Figure 6.1 presents the degree distributions of both the large and small graphs in a log-log plot. The network has a highly skewed node degree distribution, as is the case in pretty much all other networks. Applying the MLE for power law fit we obtain 1.82572 and 1.90602 respectively for the exponents. However, the quality of the fit is poor; see [38, Chapter 4] for details. The average degree of the large graph is about 3.5. The induced directed and undirected graphs in this case have both average

Table 6.1: Number of vertices and average degree of undirected subgraphs; positive polarity only, self-loops are neglected.

coreness	$\geqslant 0$	$\geqslant 2$	$\geqslant 5$	$\geqslant 8$	$\geqslant 11$	$\geqslant 14$	$\geqslant 17$	$\geqslant 20$	$\geqslant 23$	$\geqslant 26$
vertices	279,497	41,659	11,483	6,750	4,634	3,407	2,617	2,007	1,514	869
avg. degree	2.872	9.682	22.421	30.093	35.839	40.278	43.515	45.984	47.384	47.241

degrees about 3.0. The average degree of the small graph is about 40.7 while the induced directed and undirected graphs have respectively average degrees of about 16.0 and 15.1.

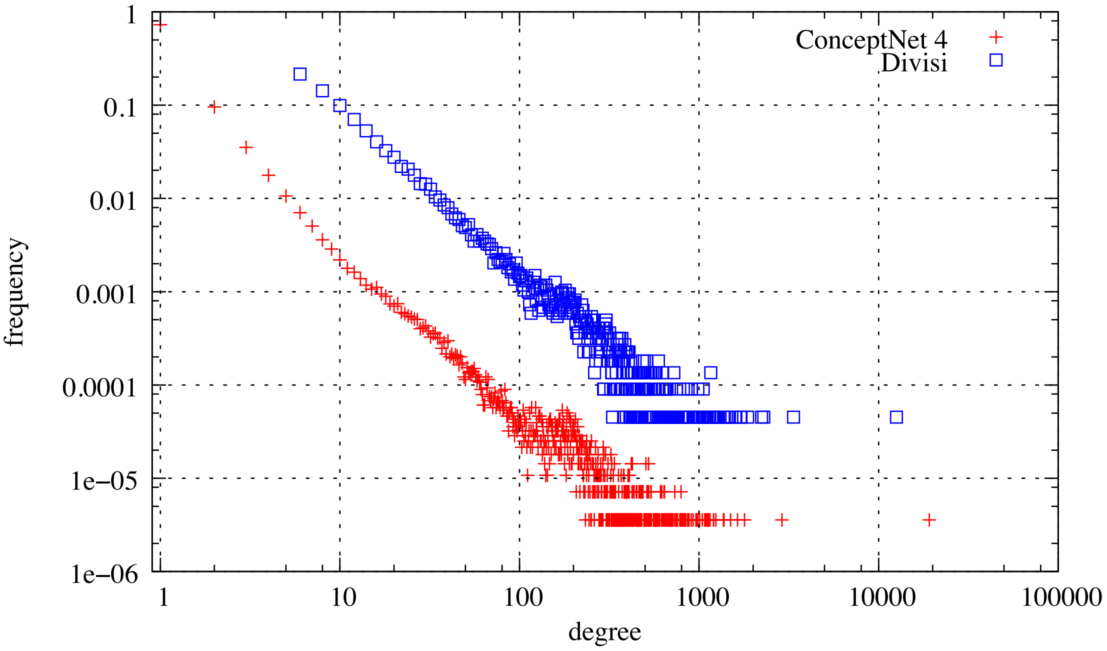


Figure 6.1: Total degree distribution in ConceptNet 4 and Divisi.

Large networks typically have a *giant component*. For the directed graph induced by assertions with any polarity, there is a giant component with 228,784 vertices, and the remaining 32,701 connected components have size at most 55, including 16,922 singletons. For *strongly connected components*, there is a giant component with 14,025 vertices, and the remaining 265,373 components have size at most 3, including 265,276 singletons.

The maximal distance in the undirected graph induced by assertions with both polarities is 16. The pair returned by `igraph` with distance 16 is `anti-charm quark` and `double-breasted de fursac jacket` [2]. The average distance in the giant component is 4.28. Thus the graph exhibits a small-world property. Details for the distances are given in [38].

The k-*core* of a graph is obtained by iteratively removing vertices of degree less than k while such vertices exist [117]. In each step there may be several choices, but it turns out that the final result is independent of those choices. The *maximum coreness* of a graph is the largest k for which the k-core is nonempty. The maximum coreness of the graph induced by the assertions with positive polarity is 26 and there are 869 concepts belonging to that core. Table 6.1 gives data on the core structure.

Now we turn to cliques, i.e., complete subgraphs, which are the strongest possible form of community

[2] Google's only reference for this concept is to ConceptNet, so this may already be an instance of an AI system creating concepts.

[3]. There are $107,100$ cliques with positive polarity, out of which there is only one clique of size 12, composed of the concepts `person`, `build`, `house`, `home`, `apartment`, `room`, `live room`, `couch`, `table`, `chair`, `cat`, and `dog`. [4] There are also 23 cliques of size 11. It turns out that all these 24 cliques are created from 36 concepts. Table 6.2 shows some of those cliques. Examining the *overlap* of cliques is also interesting as it can uncover different meanings of a concept (see also [104]) and other useful relationships. Figure 6.2 gives an example of two overlapping communities corresponding to different meanings of the concept `cut`.

Table 6.2: Concepts participating in maximal cliques with positive polarity and frequency in the range $\{5, \ldots, 10\}$. The cliques are obtained from English-language assertions with positive score. The first clique has size 12. Among all cliques of size 11 or 12 we show those where the concept `apartment` appears.

	clique								
concept	1	2	3	4	5	6	7	8	✓
apartment	✓	✓	✓	✓	✓	✓	✓	✓	8
bed		✓							1
bedroom			✓	✓	✓				3
build	✓	✓	✓	✓	✓	✓	✓	✓	8
cat	✓	✓	✓	✓		✓			5
chair	✓		✓	✓	✓	✓	✓		6
city						✓	✓	✓	3
couch	✓	✓							2
dog	✓		✓	✓	✓	✓	✓	✓	7
home	✓	✓	✓	✓	✓	✓	✓	✓	8
house	✓	✓	✓	✓	✓	✓	✓	✓	8
human					✓		✓	✓	3
live room	✓	✓							2
person	✓	✓	✓	✓	✓	✓	✓	✓	8
room	✓	✓	✓	✓	✓	✓	✓	✓	8
table	✓	✓	✓						3
town				✓	✓	✓	✓	✓	5

For *community-finding*, the *clique-percolation* algorithm [34, 104] produced interesting results. Let S be a k-clique. Clique percolation with parameter k builds a community starting from clique S and taking the union of all cliques reachable by k-chains from S, where a k-chain is a sequence of k-cliques such that each clique has $k-1$ vertices in common with the previous one [5] We found the following communities: 362 using triangles, 290 using K_4's (cliques of size 4), 287 using K_5's, 209 using K_6's, 120 using K_7's, 84 using K_8's, 16 using K_9's, 12 using K_{10}'s, 6 using K_{11}'s, and of course one community by percolating K_{12}'s. Figures 6.3 and 6.4 present communities that occur by percolating cliques of various sizes. Communities could be presented to the user for suggesting a new concept or link. For example, the concept `dishonesty` could be suggested for the community shown in Figure 6.3, resulting in the addition of new assertions. Figure 6.4 already contains `religion`, but the user might suggest the addition of an assertion involving `belief` and `prayer`.

[3] Network analysis literature uses the terms community, cluster and module interchangeably.

[4] The interpretation (*surface form*) of ConceptNet's `live room` is `living room`, or `in a living room`, etc., and of `build` is `a building`.

[5] The edge set of a community is the union of the edge sets of the cliques involved. This is not an induced subgraph in general; there can be nested communities as well.

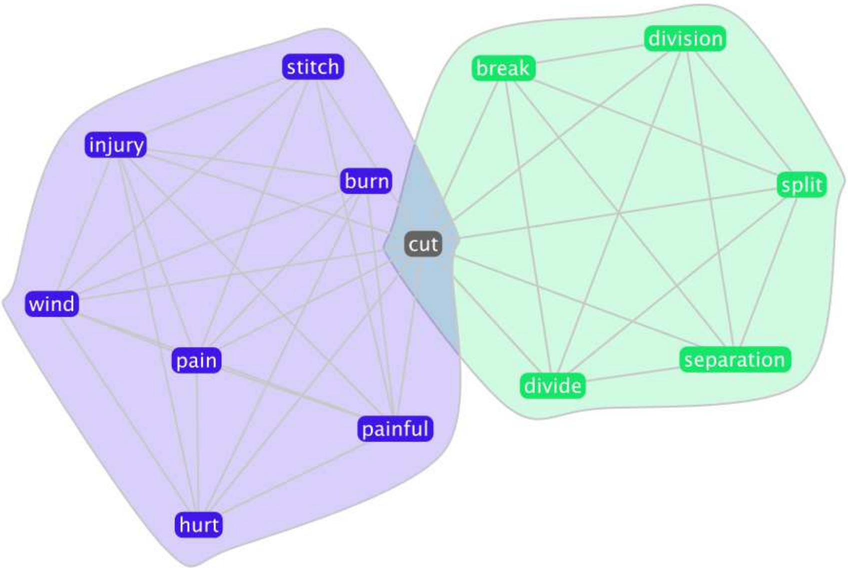


Figure 6.2: Overlapping communities for different meanings of `cut`.

6.5 Spreading Activation

Spreading activation is a technique inspired by neural models, for identifying related nodes [25], used for example, in information retrieval [27]. It is related to PageRank [103] and other similar algorithms. Recently, it has also been used in knowledge network acquisition [64]. We illustrate the application of spreading activation for query answering in the case of Comprehension queries, using a variant of Harrington's approach. Refined versions of such algorithms may be useful for improving the quality of the answers obtained.

Spreading activation can be started by activating several concepts which simultaneously spread activation values in rounds to their neighbors. The nodes also propagate their labels to neighboring nodes. The firing thresholds of the nodes and the decay factors are parameters of the process. We implemented different versions depending on the type of the underlying graph, the firing regime and the termination criterion. After the activation process is terminated, we find paths with a significant amount of activation; again, we implemented different algorithms for finding such paths. One may then start a second round of spreading activation from nodes on the significant paths, and finally look for assertions with the highest levels of activation along those paths.

We illustrate the application of spreading activation on the question Why do we put on sunscreen is summer? This turned out to be a difficult question for ConceptNet. More precisely, it turned out to be a difficult question *for the AnalogySpace-based algorithm* used in [101]. The answers received included `UsedFor/cook` and `Causes/strike match` with large weight. As we will see below, ConceptNet in fact contains sufficient information to answer the question correctly, the problem is 'only' how to find it. We also get an answer to the unexpected appearance of `cook`.

Running spreading activation from the concepts `put sunscreen` and `summer`, the first phase activates $6,700$ concepts and the three intermediate nodes where both labels appear are `heat`, `hot`, and `fall`. The undirected primary paths, involving 7 different nodes, are:

- `put sunscreen` — `go swim` — `heat` — `summer`,
- `put sunscreen` — `go swim` — `hot` — `summer`,

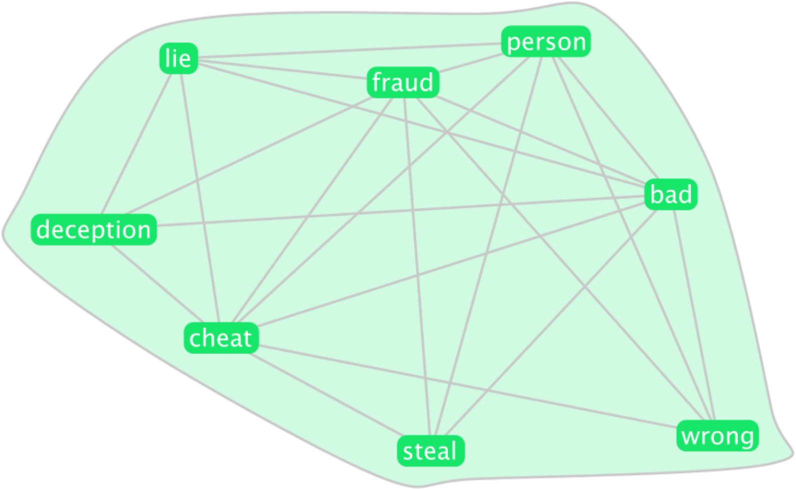


Figure 6.3: Community 'dishonest/dishonesty'. Eight nodes by percolating cliques of size 5; dishonest/dishonesty itself is missing from the community.

- put sunscreen — go fish — fall — summer.

The top ten most activated nodes in the network in the first round are, in that order, summer, put sunscreen, heat, season, hot, winter, hot weather, after spring, spring, and camp. The top ten most activated nodes in the network after the second round are, in that order, summer, heat, put sunscreen, hot, fall, go swim, go fish, fire, person, and winter.

The top ten activated pairs of concepts [6] after the first round involve the concepts summer, heat, season, hot, winter, hot weather, after spring, spring, camp, warm season, and hot month. After the second round, the concepts go swim, go fish, and put sunscreen appear in the corresponding list. In particular, the assertion (go swim, HasPrerequisite, put sunscreen) ranks in the 30th place.

Now we make some more observations on specific relations, which are relevant in the context of 'why' questions. Let us start with the relation HasPrerequisite. In both rounds the top two most activated assertions with this relation connect the pairs of concepts (go swim, put sunscreen) and (go fish, put sunscreen).

Regarding the relation CausesDesire, at the end of the first round the top three assertions in this relation connect the concepts (summer, play baseball), (summer, fish), and (summer, go walk). After the second round, the assertions (heat, CausesDesire, go swim) and (hot, CausesDesire, go swim) move to the top two positions, from positions four and five for this relation. So, we can build a slightly better justification since heat causes desire to go swimming, and going for swimming has as prerequisite to put on sunscreen.

Finally, for the Causes relation, in both rounds the top three assertions for this relation connect the pair of concepts (heat, fire), (fire, heat), and (sun, heat).

Examining other relations shows how some of the incorrect answers received in [101] are caused by multiple meanings. For example, in the first round of the spreading activation process, the top assertion for the relation HasLastSubevent connects cook meal and season, while in the end of the second round it connects climb to fall. Apparently, the problem arises due to a lack of disambiguation for

[6] A pair of concepts typically involves several assertions due to different relations connecting the concepts.

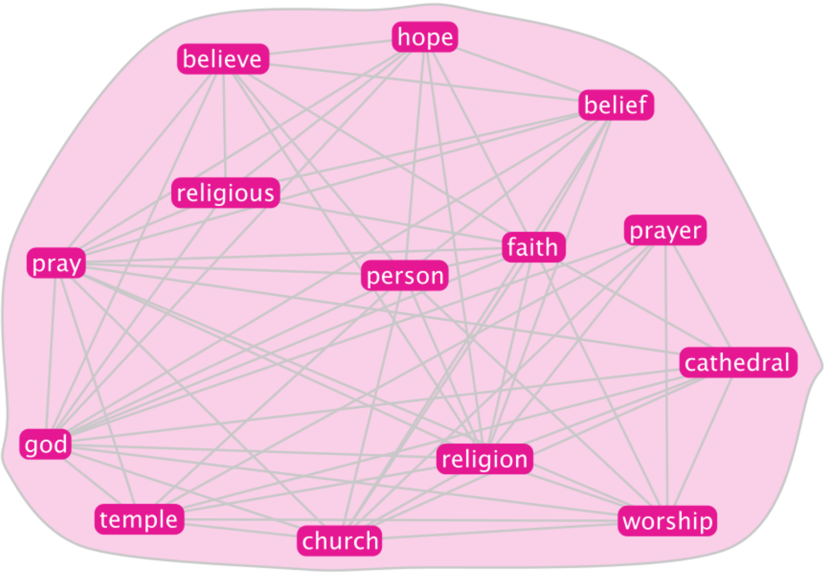


Figure 6.4: Community 'religion'. Fourteen nodes by percolating cliques of size 7; missing link between `belief` and `prayer`.

the concepts `season` and `fall`, both of which should be used here as three-month periods. However, `season` is used as the act of putting seasonings while cooking meals and `fall` is used as the verb *"to fall"*. Moreover, the idea of sunscreen in the summer typically activates nodes that are related to heat and water, which in combination with seasonings further justifies why cooking meals appears. Finally, the problem remains in the second round but due to `fall` that appears along a primary path.

6.6 Rule Mining

In this section we discuss the application of data mining towards the automated construction of a background theory for the relations used in the knowledge base. We consider rules of the simplest form, mainly for computational considerations.

A *rule* is given by an ordered triple of relations (`X`, `Y`, `Z`), where `X`, `Y` are the *premisses* and `Z` is the *conclusion*. For such a triple we consider triples of concepts (`a`, `b`, `c`) such that the assertions

(`a`, `X`, `b`) and (`b`, `Y`, `c`)

are in the knowledge base. Such triples form the *support* of the rule. If (`a`, `Z`, `c`) is also in the knowledge base then (`a`, `b`, `c`) is a *success* for the rule (`X`, `Y`, `Z`), otherwise it is a *failure*. The *success rate* of a rule is the percentage of successes in the support. Consider, for example, the rule (`Desires`, `LocatedNear`, `AtLocation`) and the triple of concepts (`human`, `drink`, `bar`). The assertions (`human`, `Desires`, `drink`) and (`drink`, `LocatedNear`, `bar`) are both in the knowledge base. Therefore, we check whether the assertion (`human`, `AtLocation`, `bar`) is in the knowledge base. It is, so (`human`, `drink`, `bar`) is a success for the rule (`Desires`, `LocatedNear`, `AtLocation`).

A triple of concepts (`a`, `b`, `c`) is *valid* for a rule (`X`, `Y`, `Z`) if the claim

(`a`, `X`, `b`) and (`b`, `Y`, `c`) therefore (`a`, `Z`, `c`)

makes sense as a reasoning step. Otherwise (`a`, `b`, `c`) is *invalid*. *Making sense is a subjective judgement* and its intended meaning is up for discussion. In what follows we use the sense "given that the premisses hold it is reasonable to assume that the conclusion holds". For example, (human, drink, bar) is valid for the rule (Desires, LocatedNear, AtLocation). Note that by the nature of its definition, deciding about validity requires an (often ambiguous) decision by a human and so computing precise statistics about it is difficult.

We performed an exhaustive test for all possible rules involving relations that have at least 300 assertions with positive score regardless of their polarity. We searched for *frequent* rules, with support at least 300 and success rate at least 5% [7]. Success rates are expected to be low even for correct rules due to the sparsity of the network. There are 76 such triples of relations. We give examples of some such relations, plus an interesting one with low success rate, and comment on issues raised by these examples.

Our first example is the rule (Desires, LocatedNear, AtLocation). This is the highest scoring rule with 251 successes and support 2050 (12% success rate). The triples (human, drink, bar) and (bird, seed, garden) are successful and valid. The triple (human, love, heart) is successful but invalid. The triple (bird, seed, plant garden) is a failure but it is valid. The reason for the failure is that the assertion (bird, AtLocation, plant garden) is missing from the knowledge base. This is an example of using the mined rules to identify missing entries.

The rule (AtLocation, PartOf, AtLocation) has 2,394 successes and support 27,917 (8.5% success rate). The triple (text book, classroom, school) is successful and valid. On the other hand, (text book, classroom, school system) is a failure. In contrast to the failure discussed for the first rule above, this is not due to a missing assertion, because the triple is *invalid*. This points to a general problem with this rule: it is only expected to hold if the third concept is a physical object, like school and unlike school system. Thus examining this example suggests a weakening of the rule.

The rule (PartOf, AtLocation, AtLocation) is similar to the previous one. However, its success rate is much smaller, only 1.4% (with support 78,804, but only 1,112 successes). A possible explanation of the discrepancy can be illustrated by the triple (engine oil, car, town). It is a failure as the assertion (engine oil, AtLocation, town) is not in the knowledge base. Its validity depends on the status of (engine oil, AtLocation, town). This assertion is not to be expected as input from a user (or from a text). On the other hand, it is reasonable as a factual statement about the world.

Let us elaborate on the difference between the two rules. For (AtLocation, PartOf, AtLocation), the combined facts that a is an appropriate[8] left argument for AtLocation, b is an appropriate right argument for AtLocation, and (b, PartOf, c) mean that if c is an appropriate right argument for AtLocation (like school but unlike school system) then the assertion (a, AtLocation, c) *makes sense both as a factual statement about the world and in terms of natural language usage*. By way of contrast, for (PartOf, AtLocation, AtLocation), things that are appropriate as left arguments for PartOf are normally not thought of as appropriate left arguments for AtLocation; if they do occur as such a left argument then they occur as being AtLocation of the thing they are part of. Thus, in this case (a, AtLocation, c) *may make sense as a factual statement about the world but not in terms of natural language usage*. Thus, the observed difference between the success rates of two similar rules points to a *possible mismatch between natural language usage and intended question answering applications*. This may be an issue to consider for further knowledge base development.

The rule (LocatedNear, PartOf, IsA) does not make much sense even if it has 253 successes and support 4252 (6% success rate). Most successes we examined are false or nonsensical. This is an example of a rule with high success rate but with many successful, invalid triples. An example is the triple (desk, classroom, school). The wrong assertion (desk, IsA, school) comes from the sentence Schools have desks through the intermediate form Desk is a type of school. Thus the problem

[7] For rules involving more than three concepts such an exhaustive search is not feasible, and it will be necessary to use more advanced data mining techniques.

[8] By *appropriate* we mean "makes common sense for users asked to give natural language statements".

presumably comes from a programming error and fixing it might eliminate many wrong assertions. Hence this in an example where rule mining can be used to correct mistakes.

Chapter 7

Conclusion

In the case of evolvability, we saw that even simple evolutionary mechanisms like the swapping algorithm for monotone conjunctions, pose interesting questions about the learnability of well-understood concept classes. We consider the study of monotone conjunctions, even under specific distributions, the first step on a tour of exploration of more Boolean functions through evolutionary mechanisms that comply with similarly intuitive neighborhoods. For each one of these Boolean functions we can think of two "natural" mutations for the neighborhoods in terms of proper learning; manipulating variables or manipulating the Fourier coefficients for their natural representation in a spirit similar to Michael's for 1-decision lists [94]. Moreover, we have not yet seen an algorithm that uses a random-walk type of argument, instead of the strictly beneficial steps followed until convergence. Perhaps this would be an important breakthrough. Working with different fitness functions is another promising avenue as it is suggested by using covariance in Chapter 3. Finally, Kanade in [72] extended the evolvability framework by allowing recombination. However, this interesting variant of evolvability is virtually unexplored.

Regarding MIL, the first idea for extending our work is the study of the learnability of Boolean functions under this particular setting. For instance what is the exact VC dimension of conjunctions under MIL? What about the VC dimension of other Boolean concept classes studied under MIL? What about specific algorithms tailored for specific Boolean concept classes under the MIL setting?

Another interesting direction, which was actually our original motivation for studying both MIL as well as AL, is the combined setup of multiple-instance active learning (MIAL) setting. This particular setting (MIAL) has received some attention in machine learning, but, as far as we know, has not been considered so far in learning theory. We point out the applicability of some recent active learning results in the context of multi-instance learning, without giving a detailed definition of the notions involved.

There are several possibilities for formulating a model of active learning in the multi-instance model, such as querying bag labels, or various ways of querying instance labels within bags, and these variants may be relevant in different learning scenarios (see Settles, Craven, and Ray [121]). Here we assume that the learner gets unlabeled r-bags and then is charged for querying the label of a bag.

The *mellow algorithm* which was discussed in Section 5.1 queries the label of a bag iff its label is not determined by the labels of the previously queried bags. Hanneke's bound for the analysis of the label complexity of the mellow algorithm has linear dependence with respect to the disagreement coefficient ρ; see Theorem 5.2.4. Hence, when the disagreement coefficient has at most logarithmic dependence on ε, Hanneke's bound implies that the mellow algorithm achieves an exponential speedup compared to traditional supervised learning. Friedman [49] proved a general bound for the disagreement coefficient. In particular, his results, and therefore Hanneke's bounds, apply to the learning of hyperplanes over *smooth* distributions. Friedman assumes a smoothness condition for the combined parametrized representation of instances and concepts, but he also gives several extensions to cases where such assumptions do not hold. Multi-instance learning of r-bags of d-dimensional halfspaces corresponds to learning concepts in (rd)-dimensional space of the form

$$\{(x_1^1,\ldots,x_d^1,\ldots,x_1^r,\ldots,x_d^r) : w_1x_1^i+\ldots+w_dx_d^i \geqslant t \text{ for some } i,\ 1\leqslant i\leqslant i\}.$$

Among the extensions discussed by Friedman, this class is covered by, for example, the result of Balcan, Hanneke, and Wortman [10] on the union of exponential rate classes. Thus we conclude that halfspaces are *actively learnable from bags at an exponential rate*.

The mellow algorithm for active learning has an efficient implementation whenever hypothesis finding can be done efficiently; see Algorithm 3 in Section 5.1. A new instance has to be queried iff the previously queried labels are consistent with both labels for the new instance. This, again, does not work for bags of halfspaces. Thus it seems to be an open problem whether there is an efficient active learning algorithm with exponential error rate.

Ultimately we would like to study Boolean concept classes under the joint MIAL framework.

Regarding active learning specifically, we consider the analysis of the disagreement coefficient of monotone conjunctions under the uniform distribution $\mathcal{U}_n$ in Chapter 5 as the first step on an exploratory journey of Boolean concept classes under the AL framework. In particular, as the accuracy ε decreases over time, is there a refined analysis of Hanneke's theorem for the mellow algorithm (see Theorem 5.2.4) that is based on the study of the disagreement region which was presented in Chapter 5? As a reminder, targets of size k such that $2\leqslant k\leqslant\lfloor n/2\rfloor$ have a disagreement coefficient that is $\Theta\left(2^k\right)$. Requiring accuracy $\Theta\left(2^{-k}\right)$ implies that we have to identify these targets precisely and that the disagreement coefficient is $\Theta\left(1/\varepsilon\right)$. As a consequence, the current bound of Hanneke in Theorem 5.2.4 does not imply an improvement over the traditional supervised learning setup. To see this, first note that the VC dimension of monotone conjunctions is n; see for example [98]. By Theorem 2.5.4, traditional supervised learning, for *arbitrary* distributions, requires $\Omega\left(d/\varepsilon\right)=\Omega\left(n/\varepsilon\right)=\Omega\left(n\cdot 2^{\alpha\cdot n}\right)$ examples in order to identify such a target precisely. Hence, an improved bound on the label complexity of the mellow algorithm that is based on the complete study of the disagreement region for monotone conjunctions under $\mathcal{U}_n$ would be an interesting new result in AL.

However, it appears that even if such an analysis does not yield a better bound, one can actually hope for a better bound through a different route. The teaching dimension [55] and in particular the extended teaching dimension considered by Hegedüs in [68] appears to be a promising avenue of research[1]. Letting d denote the extended teaching dimension of a concept class $\mathcal{C}$, Hegedüs has proved in [68] that the number of membership queries required for concept learning a target $c\in\mathcal{C}$ is bounded from above by $\mathcal{O}\left(d\cdot\lg|\mathcal{C}|\right)$. Hence, studying the different variations of the *halving algorithm* [68] in the framework of AL, just like Hegedüs did in the framework of the membership query model, seems to be an important alternative for the analysis of AL; see [61].

In the case of knowledge bases and reasoning, in Chapter 6, we considered the ConceptNet knowledge base from the point of view of network analysis. We discussed degree distribution, small world property, cores, cliques and communities. We also discussed spreading activation and rule mining for the relations used in ConceptNet. The results that occurred through network analysis suggest possible applications to improved question answering. For example, the inclusion of communities in order to find missing assertions, using spreading activation to find answers and explanations for *why*-questions, and using rule mining to find missing assertions and correct errors.

The mined rules, such as the transitivity of IsA, could be used to add many new assertions. However, adding all these assertions is neither feasible, nor desirable, as it would make the knowledge base denser. The rules appear to be more useful as a background theory, to be used in deriving and refining answers. This could be one instance of building additional knowledge into the system.

[1] Both of these combinatorial parameters have been studied in the traditional *membership query* model of learning [4]; see also [89].

ConceptNet provides a possibility to combine statistical and logic-based approaches to commonsense reasoning, exemplified by SVD and spreading activation. Exploring ways of combining the two approaches to enhance performance is an interesting research direction.

Finally, how much commonsense reasoning capability is implicit in a large commonsense knowledge base like ConceptNet? In other words, using the familiar metaphor, does this approach lead to the moon or is it just a tree (hopefully, at least a tall one in that case)? Of course it is too early to even guess an answer, but we hope that the explorations outlined in this paper might prove to be useful towards answering this fundamental question.

Bibliography

[1] Scott Aaronson. Why Philosophers Should Care About Computational Complexity. Available at ECCC: http://eccc.hpi-web.de/report/2011/108, August 2011.

[2] Lada Adamic and Eytan Adar. Friends and neighbors on the Web. *Social Networks*, 25(3):211–230, 2003.

[3] John R. Anderson. A spreading activation theory of memory. *Journal of Verbal Leaning and Verbal Behaviour*, 22:261–295, 1983.

[4] Dana Angluin. Queries and Concept Learning. *Machine Learning*, 2(4):319–342, 1987.

[5] Martin Anthony and Norman Biggs. *Computational Learning Theory*. Cambridge University Press, New York, NY, USA, 1997.

[6] Les E. Atlas, David A. Cohn, and Richard E. Ladner. Training Connectionist Networks with Queries and Selective Sampling. In *NIPS*, pages 566–573, 1989.

[7] Peter Auer, Philip M. Long, and Aravind Srinivasan. Approximating Hyper-Rectangles: Learning and Pseudo-Random Sets. In *STOC*, pages 314–323, 1997.

[8] Maria-Florina Balcan, Chris Berlind, Steven Ehrlich, and Yingyu Liang. Efficient Semi-Supervised and Active Learning of Disjunctions. In *ICML*, 2013. To appear.

[9] Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. *Journal of Computer and System Sciences*, 75(1):78–89, January 2009.

[10] Maria-Florina Balcan, Steve Hanneke, and Jennifer Wortman. The True Sample Complexity of Active Learning. In *COLT*, pages 45–56, 2008.

[11] Marcello Balduccini, Chitta Baral, and Yuliya Lierler. Knowledge representation and question answering. In *Handbook of Knowledge Representation*, pages 779–819. Elsevier, 2008.

[12] Tanya Berger-Wolf, Dimitrios I. Diochnos, András London, András Pluhár, Robert H. Sloan, and György Turán. Commonsense knowledge bases and network analysis. In *Commonsense*, May 2013.

[13] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008+, Jul 2008. Also available at arXiv:0803.0476 [physics.soc-ph].

[14] Avrim Blum and Adam Kalai. A Note on Learning from Multiple-Instance Examples. *Machine Learning*, 30(1):23–29, January 1998.

[15] Avrim Blum and Adam Kalai. A Note on Learning from Multiple-Instance Examples. *Machine Learning*, 30(1):23–29, January 1998.

[16] Anselm Blumer, A. Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, October 1989.

[17] Ronald J. Brachman and Hector J. Levesque. *Knowledge Representation and Reasoning.* Elsevier, 2004.

[18] U. Brandes and T. Erlebach. *Network Analysis: Methodological Foundations*, volume 3418 of *LNCS*. Springer, 2005.

[19] Nader H. Bshouty and Vitaly Feldman. On Using Extended Statistical Queries to Avoid Membership Queries. *Journal of Machine Learning Research*, 2:359–395, 2002.

[20] Jorge Castro and José L. Balcázar. Simple PAC Learning of Simple Decision Lists. In *ALT '95: Proceedings of the 6th International Conference on Algorithmic Learning Theory*, pages 239–248, London, UK, 1995. Springer-Verlag.

[21] Herman Chernoff. A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on the sum of Observations. *Annals of Mathematical Statistics*, 23:409–507, 1952.

[22] Aaron Clauset, Mark E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, 70(6):066111+, Dec 2004. Also available at arXiv:cond-mat/0408187 [cond-mat.stat-mech].

[23] Aaron Clauset, Cosma Rohilla Shalizi, and Mark E. J. Newman. Power-Law Distributions in Empirical Data. *SIAM Review*, 51(4):661–703, November 2009.

[24] David A. Cohn, Les E. Atlas, and Richard E. Ladner. Improving Generalization with Active Learning. *Machine Learning*, 15(2):201–221, 1994.

[25] A. M. Collins and E. F. Loftus. A spreading-activation theory of semantic processing. *Psychological Review*, 82(6):407, 1975.

[26] Raul Cordovil and Pierre Duchet. Cyclic Polytopes and Oriented Matroids. *Eur. J. Comb.*, 21(1):49–64, 2000.

[27] Fabio Crestani. Application of Spreading Activation Techniques in Information Retrieval. *Artificial Intelligence Review*, 11(6):453–482, 1997.

[28] Gábor Csárdi and Tamás Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.

[29] Ido Dagan and Sean P. Engelson. Committee-Based Sampling For Training Probabilistic Classifiers. In *ICML*, pages 150–157, 1995.

[30] Charles Darwin. *On the origin of species by means of natural selection.* London: John Murray, 1859.

[31] Sanjoy Dasgupta. Two faces of active learning. *Theor. Comput. Sci.*, 412(19):1767–1781, 2011.

[32] Sanjoy Dasgupta, Daniel Hsu, and Claire Monteleoni. A general agnostic active learning algorithm. In *NIPS*, 2007.

[33] Simon De Deyne and Gert Storms. Word associations: Network and semantic properties. *Behavior Research Methods*, 40(1):213–231, 2008.

[34] Imre Derényi, Gergely Palla, and Tamás Vicsek. Clique percolation in random networks. *Physical review letters*, 94(16):160202, 2005.

[35] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition.* Springer, 1996.

[36] Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the Multiple Instance Problem with Axis-Parallel Rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997.

[37] Dimitrios I. Diochnos. Leveling-Up in Heroes of Might and Magic III. In Paolo Boldi and Luisa Gargano, editors, *FUN*, pages 145–155. Springer, 2010.

[38] Dimitrios I. Diochnos. Commonsense Reasoning and Large Network Analysis: A Computational Study of ConceptNet 4. *arXiv:abs/1304.5863 [cs/AI]*, 2013.

[39] Dimitrios I. Diochnos, Robert H. Sloan, and György Turán. On multiple-instance learning of halfspaces. *Information Processing Letters*, 112(23):933–936, December 2012.

[40] Dimitrios I. Diochnos and György Turán. On Evolvability: The Swapping Algorithm, Product Distributions, and Covariance. In Osamu Watanabe and Thomas Zeugmann, editors, *SAGA*, pages 74–88. Springer, 2009.

[41] Dimitris Diochnos. Application of Reinforcement Learning and Combinatorial Search to One-Player Games. Undergraduate thesis, Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Hellas, February 2004.

[42] Dimitris Diochnos. Real Solving on Algebraic Systems of Small Dimension. Master's thesis, Department of Mathematics, National and Kapodistrian University of Athens, Hellas, June 2007.

[43] Dimitris Diochnos. Solving Algebraic Systems of Small Dimension over the Reals. *Selected Undergraduate and Graduate Theses, Department of Informatics and Telecommunications, National and Kapodistrian University of Athens*, 2008(9):23–32, 2008.

[44] David Easley and Jon Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World.* Cambridge University Press, 2010.

[45] Andrzej Ehrenfeucht, David Haussler, Michael Kearns, and Leslie Valiant. A General Lower Bound on the Number of Examples Needed for Learning. *Information and Computation*, 82(3):247–261, 1989.

[46] Vitaly Feldman. Evolvability from learning algorithms. In *STOC '08: Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 619–628, New York, NY, USA, 2008. ACM.

[47] Vitaly Feldman. Robustness of Evolvability. In *COLT 2009: Conference on Learning Theory*, 2009.

[48] Vitaly Feldman and Leslie G. Valiant. The Learning Power of Evolution. In Rocco A. Servedio and Tong Zhang, editors, *COLT*, pages 513–514. Omnipress, 2008.

[49] Eric J. Friedman. Active Learning for Smooth Problems. In *COLT*, 2009.

[50] Atsushi Fujii, Kentaro Inui, Takenobu Tokunaga, and Hozumi Tanaka. Selective Sampling for Example-based Word Sense Disambiguation. *Computational Linguistics*, 24(4):573–597, 1998.

[51] Merrick L. Furst, Jeffrey C. Jackson, and Sean W. Smith. Improved learning of AC0 functions. In *COLT '91: Proceedings of the fourth annual workshop on Computational learning theory*, pages 317–325, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc.

[52] Matt Ginsberg. *Essentials of artificial intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1994.

[53] Michelle Girvan and Mark E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[54] E. Mark Gold. Complexity of Automaton Identification from Given Data. *Information and Control*, 37(3):302–320, 1978.

[55] Sally A. Goldman and Michael J. Kearns. On the Complexity of Teaching. In *COLT*, pages 303–314, 1991.

[56] M. Gonen, D. Ron, U. Weinsberg, and A. Wool. Finding a dense-core in Jellyfish graphs. *Computer Networks*, 52(15):2831–2841, 2008.

[57] Ronald Graham, Bruce Rothschild, and Joel H. Spencer. *Ramsey Theory*. John Wiley & Sons Inc., second edition, 1990.

[58] Torben Hagerup and C. Rüb. A guided tour of Chernoff bounds. *Information Processing Letters*, 33(6):305–308, February 1990.

[59] Thomas Hancock and Yishay Mansour. Learning monotone ku DNF formulas on product distributions. In *COLT '91: Proceedings of the fourth annual workshop on Computational learning theory*, pages 179–183, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc.

[60] Steve Hanneke. A bound on the label complexity of agnostic active learning. In *ICML*, pages 353–360, 2007.

[61] Steve Hanneke. Teaching dimension and the complexity of active learning. In *Proceedings of the 20th annual conference on Learning theory*, COLT'07, pages 66–81, Berlin, Heidelberg, 2007. Springer-Verlag.

[62] Steve Hanneke. *Theoretical Foundations of Active Learning*. PhD thesis, Carnegie Mellon University, Machine Learning Department, 2009.

[63] Steve Hanneke. Activized Learning: Transforming Passive to Active with Improved Label Complexity. *Journal of Machine Learning Research*, 13(5):1469–1587, 2012.

[64] B. Harrington and S. Clark. Asknet: Automated semantic knowledge network. In *AAAI*, 2007.

[65] Alexander G. Hauptmann, Wei-Hao Lin, Rong Yan, Jun Yang, and Ming yu Chen. Extreme video retrieval: joint maximization of human and computer performance. In *ACM Multimedia*, pages 385–394, 2006.

[66] C. Havasi, J. Pustejovsky, R. Speer, and H. Lieberman. Digital intuition: Applying common sense using dimensionality reduction. *Intelligent Systems, IEEE*, 24(4):24–35, 2009.

[67] C. Havasi, R. Speer, and J. Alonso. ConceptNet 3: a flexible, multilingual semantic network for common sense knowledge. In *Recent Advances in Natural Language Processing*, pages 27–29, 2007.

[68] Tibor Hegedüs. Generalized Teaching Dimensions and the Query Complexity of Learning. In *COLT*, pages 108–117, 1995.

[69] Wassily Hoeffding. Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, 58(301):13–30, March 1963.

[70] Steven C. H. Hoi, Rong Jin, and Michael R. Lyu. Large-scale text categorization by batch mode active learning. In *WWW*, pages 633–642, 2006.

[71] Adam Tauman Kalai and Shang-Hua Teng. Decision trees are PAC-learnable from most product distributions: a smoothed analysis. *CoRR*, abs/0812.0933, 2008. Informal publication.

[72] Varun Kanade. Evolution with Recombination. In *FOCS*, pages 837–846, 2011.

[73] Varun Kanade, Leslie G. Valiant, and Jennifer Wortman Vaughan. Evolution with Drifting Targets. In *COLT*, pages 155–167, 2010.

[74] Michael Kearns. Efficient noise-tolerant learning from statistical queries. *J. ACM*, 45(6):983–1006, 1998.

[75] Michael J. Kearns, Robert E. Schapire, and Linda Sellie. Toward Efficient Agnostic Learning. *Machine Learning*, 17(2-3):115–141, 1994.

[76] Michael J. Kearns and Umesh V. Vazirani. *An introduction to computational learning theory.* MIT Press, Cambridge, MA, USA, 1994.

[77] John George Kemeny and James Laurie Snell. *Finite Markov Chains.* Van Nostrand, New York, 1960.

[78] Donald E. Knuth. Computer Science and Its Relation to Mathematics. *The American Mathematical Monthly*, 81:323–343, 1974.

[79] Alexander Kotov and ChengXiang Zhai. Tapping into knowledge base for concept feedback: leveraging conceptnet to improve search results for difficult queries. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 403–412, 2012.

[80] Vikram Krishnamurthy. Algorithms for optimal scheduling and management of hidden Markov model sensors. *IEEE Transactions on Signal Processing*, 50(6):1382–1397, 2002.

[81] O. Erhun Kundakcioglu, Onur Seref, and Panos M. Pardalos. Multiple instance learning via margin maximization. *Applied Numerical Mathematics*, 60(4):358–369, April 2010.

[82] K. J. Lang and E. B. Baum. Query learning can work poorly when a human oracle is used. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, pages 335–340, 1992.

[83] Douglas B. Lenat. CYC: A Large-Scale Investment in Knowledge Infrastructure. *Communications of the ACM*, 38(11):32–38, 1995.

[84] Douglas B. Lenat and R. V. Guha. *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, first edition, 1989.

[85] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009.

[86] David D. Lewis and William A. Gale. A Sequential Algorithm for Training Text Classifiers. In *SIGIR*, pages 3–12, 1994.

[87] Hugo Liu and Push Singh. ConceptNet – A Practical Commonsense Reasoning Tool-Kit. *BT Technology Journal*, 22:211–226, 2004.

[88] Ying Liu. Active learning with support vector machine applied to gene expression data for cancer classification. *Journal of Chemistry Information and Computer Science*, 44:1936–1941, 2004.

[89] Wolfgang Maass and György Turán. Lower bound methods and separation results for on-line learning models. *Machine Learning*, 9(2-3):107–145, 1992.

[90] Jirí Matousek. *Lectures on Discrete Geometry.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.

[91] Mark T. Maybury. Question answering: An introduction. In *New directions in question answering*, pages 3–8. AAAI Press, November 2004.

[92] Andrew McCallum and Kamal Nigam. Employing EM and Pool-Based Active Learning for Text Classification. In *ICML*, pages 350–358, 1998.

[93] J. McCarthy. Programs with common sense. In *Proceedings of the Teddington Conference on the Mechanization of Thought Processes*, pages 756–91, 1959.

[94] Loizos Michael. Evolvability via the Fourier transform. *Theoretical Computer Science*, 462:88–98, 2012.

[95] Tom M. Mitchell. Generalization as Search. *Artificial Intelligence*, 18(2):203–226, 1982.

[96] Tom M. Mitchell. *Machine learning.* McGraw Hill series in computer science. McGraw-Hill, 1997.

[97] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms.* Cambridge University Press, 1995. 8th printing, 2006.

[98] Thomas Natschläger and Michael Schmitt. Exact VC-Dimension of Boolean Monomials. *Information Processing Letters*, 59(1):19–20, 1996.

[99] Mark E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

[100] Mark E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74:036104, Sep 2006. Also available at arXiv:physics/0605087 [physics.data-an].

[101] Stellan Ohlsson, Robert H. Sloan, György Turán, Daniel Uber, and Aaron Urasky. An Approach to Evaluate AI Commonsense Reasoning Systems. In *FLAIRS Conference*, pages 371–374, 2012.

[102] Stellan Ohlsson, Robert H. Sloan, György Turán, and Aaron Urasky. The ConceptNet 4 artificial intelligence system has the verbal IQ of a four-year-old. Submitted for publication., 2013.

[103] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[104] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, June 2005.

[105] Christos H. Papadimitriou. *Computational complexity.* Addison-Wesley, 1994.

[106] Pascal Pons and Matthieu Latapy. Computing Communities in Large Networks Using Random Walks. *Journal of Graph Algorithms and Applications*, 10(2):191–218, 2006. The long version is available at arXiv:physics/0512106 [physics.soc-ph].

[107] M. Ross Quillian. The teachable language comprehender: a simulation program and theory of language. *Comm. ACM*, 12(8):459–476, Aug 1969.

[108] Usha N. Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3):036106+, Sep 2007. Also available at arXiv:0709.2938 [physics.soc-ph].

[109] Jörg Reichardt and Stefan Bornholdt. Statistical mechanics of community detection. *Physical Review E*, 74:016110, Jul 2006. Also available at arXiv:cond-mat/0603718 [cond-mat.dis-nn].

[110] Rüdiger Reischuk and Thomas Zeugmann. A Complete and Tight Average-Case Analysis of Learning Monomials. In *Proc. 16th Int'l Sympos. on Theoretical Aspects of Computer Science, STACS'99*, pages 414–423. Springer, 1999.

[111] Johannes P. Ros. Learning Boolean functions with genetic algorithms: A PAC analysis. In *Foundations of Genetic Algorithms*, pages 257–275, San Mateo, CA, 1993. Morgan Kaufmann.

[112] Martin Rosvall and Carl T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008. Also available at arXiv:0707.0609 [physics.soc-ph].

[113] Stuart J. Russell and Peter Norvig. *Artificial Intelligence - A Modern Approach.* Pearson Education, third edition, 2010.

[114] Sivan Sabato and Naftali Tishby. Homogeneous Multi-Instance Learning with Arbitrary Dependence. In *COLT*, 2009.

[115] Purnamrita Sarkar, Deepayan Chakrabarti, and Andrew W. Moore. Theoretical Justification of Popular Link Prediction Heuristics. In *IJCAI*, pages 2722–2727, 2011.

[116] Norbert Sauer. On the Density of Families of Sets. *J. Comb. Theory, Ser. A*, 13(1):145–147, 1972.

[117] S. B. Seidman. Network structure and minimum degree. *Social networks*, 5(3):269–287, 1983.

[118] Rocco A. Servedio. On learning monotone DNF under product distributions. *Inf. Comput.*, 193(1):57–74, 2004.

[119] Burr Settles. *Active Learning.* Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2012.

[120] Burr Settles and Mark Craven. An Analysis of Active Learning Strategies for Sequence Labeling Tasks. In *EMNLP*, pages 1070–1079, 2008.

[121] Burr Settles, Mark Craven, and Soumya Ray. Multiple-Instance Active Learning. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1289–1296. MIT Press, Cambridge, MA, 2008.

[122] Saharon Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41:247–261, 1972.

[123] Edward Shen, Henry Lieberman, and Francis Lam. What am I gonna wear?: Scenario-oriented recommendation. In *Proceedings of the 12th international conference on Intelligent user interfaces*, pages 365–368, 2007.

[124] Hans-Ulrich Simon. Learning decision lists and trees with equivalence-queries. In *EuroCOLT '95: Proceedings of the Second European Conference on Computational Learning Theory*, pages 322–336, London, UK, 1995. Springer-Verlag.

[125] Michael Sipser. *Introduction to the theory of computation.* PWS Publishing Company, 1997.

[126] R. Speer, K. Arnold, and C. Havasi. Divisi: Learning from Semantic Networks and Sparse SVD. In *Proc. 9th Python in Science Conf. (SCIPY 2010)*, 2010.

[127] R. Speer, C. Havasi, and H. Lieberman. AnalogySpace: reducing the dimensionality of common sense knowledge. In *AAAI*, 2008.

[128] Robert Speer, Catherine Havasi, and Henry Lieberman. AnalogySpace: Reducing the Dimensionality of Common Sense Knowledge. In *AAAI*, pages 548–553, 2008.

[129] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM*, 51(3):385–463, 2004.

[130] Mark Steyvers and Joshua B. Tenenbaum. The Large-Scale Structure of Semantic Networks: Statistical Analyses and a Model of Semantic Growth. *Cognitive Science*, 29(1):41–78, 2005.

[131] Cynthia A. Thompson, Mary Elaine Califf, and Raymond J. Mooney. Active Learning for Natural Language Parsing and Information Extraction. In *ICML*, pages 406–414, 1999.

[132] Simon Tong and Edward Y. Chang. Support vector machine active learning for image retrieval. In *ACM Multimedia*, pages 107–118, 2001.

[133] Simon Tong and Daphne Koller. Support Vector Machine Active Learning with Application sto Text Classification. In *ICML*, pages 999–1006, 2000.

[134] Gökhan Tür, Dilek Z. Hakkani-Tür, and Robert E. Schapire. Combining active and semi-supervised learning for spoken language understanding. *Speech Communication*, 45(2):171–186, 2005.

[135] Alan M. Turing. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings London Mathematical Society*, 2(42):230–265, 1936.

[136] Alan M. Turing. Computing Machinery and Intelligence. *Mind*, 59:433–460, 1950.

[137] Leslie G. Valiant. A Theory of the Learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[138] Leslie G. Valiant. Evolvability. In Ludek Kucera and Antonín Kucera, editors, *MFCS*, volume 4708 of *Lecture Notes in Computer Science*, pages 22–43. Springer, 2007.

[139] Leslie G. Valiant. Evolvability. *Journal of the ACM*, 56(1):1–21, 2009.

[140] Vladimir Vapnik. *Estimation of Dependences Based on Empirical Data: Springer Series in Statistics (Springer Series in Statistics).* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1982.

[141] Vladimir N. Vapnik and Alexey Ya. Chervonenkis. On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.

[142] Suzan Verberne, Lou Boves, Nelleke Oostdijk, and Peter-Arno Coppen. What is not in the Bag of Words for Why-QA? *Computational Linguistics*, 36(2):229–245, 2010.

[143] Ken Wakita and Toshiyuki Tsurumi. Finding Community Structure in Mega-scale Social Networks. *CoRR*, abs/cs/0702048, 2007. Available at arXiv:cs/0702048 [cs.CY].

[144] Duncan Watts and S. Strogatz. Collective dynamics of small-world networks. *Nature*, 393:440–442, 1998.

[145] Rong Yan, Jie Yang, and Alexander G. Hauptmann. Automatically Labeling Video Data Using Multi-class Active Learning. In *ICCV*, pages 516–523, 2003.

[146] Hwanjo Yu. SVM selective sampling for ranking with application to data retrieval. In *KDD*, pages 354–363, 2005.

[147] Cha Zhang and Tsuhan Chen. An active learning framework for content-based information retrieval. *IEEE Transactions on Multimedia*, 4(2):260–268, 2002.

Appendix A

Permissions

The content of Chapter 3 is based on [40]. Since the author of this thesis is also one of the authors in [40], the author of this thesis is allowed to include part or all of the content that appeared in the original article in [40]. The license numbered 3191970208570 was obtained on July 18, 2013, through the Copyright Clearance Center's RightsLink service, and is between Dimitrios Diochnos and Springer. Permission for reusing the figures that appeared in [40] was obtained through the license number 3191971334671 on July 18, 2013, through the Copyright Clearance Center's RightsLink service, and is between Dimitrios Diochnos and Springer.

The content of Chapter 4 is based on [39]. Since the author of this thesis is also one of the authors in [39], the author of this thesis is allowed to include part or all of the content that appeared in the original article in [39]. The license numbered 3191950770851 was obtained on July 18, 2013, through the Copyright Clearance Center's RightsLink service, and is between Dimitrios Diochnos and Elsevier.

The content of Chapter 6 is based on [12] as well as on [38]. Neither of these two works, that is [12] and [38] is subject to any copyrights and hence their content is used without any further permissions or licenses.

Appendix B

Vita

Scientific Interests

Randomized Algorithms, PAC Learning, Multiple-Instance Learning, Active Learning, Evolvability, Reinforcement Learning, Markov Chains and Stochastic Processes, Artificial Intelligence, Knowledge Bases, Network Analysis, Spreading Activation & Information Retrieval, Discrete Mathematics, Exact Computations, Computer Algebra and Real Solving, Theoretical Computer Science more broadly.

Education

2013 Ph.D. in Mathematical Computer Science
Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, USA

Thesis *Analysis of Algorithms in Learning Theory and Network Analysis of Knowledge Bases*

Adviser György Turán

2007 M.Sc. in Logic, Theory of Algorithms, and Computation
Interdisciplinary Graduate Program in Logic, Algorithms and Computation, Department of Mathematics, National and Kapodistrian University of Athens, Hellas

Thesis *Real Solving on Algebraic Systems of Small Dimension*

Adviser Ioannis Z. Emiris

2004 Ptychion in Informatics and Telecommunications
Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Hellas

Thesis *Application of Reinforcement Learning and Combinatorial Search to One-Player Games*

Adviser Panagiotis Stamatopoulos

Languages

Fluent Hellenic (native), English

Moderate German

Fellowships & Awards

Teaching Award MCS 260 - Introduction to Computer Science, Fall 2009

Graduate UIC Chancellor's Graduate Research Fellowship, Spring - Summer 2010-2011

Undergraduate "Zossima Brothers" foundation fellow

Scientific Activities & Service

January 2012 Publicity Chair, *Twelfth International Symposium on Artificial Intelligence and Mathematics*, ISAIM 2012, Fort Lauderdale, FL, USA

February 2010 Webmaster, *Workshop in Graph Theory and Combinatorics in Memory of Uri Peled*, University of Illinois at Chicago, Chicago, IL, USA

September 2004 Member of the International Scientific Committee (ISC) at the *International Olympiad in Informatics*, IOI-2004, Athens, Attiki, Hellas

Reviewer CASC, SODA

Employment History

Aug 2007–Aug 2013 Appointments as a Teaching Assistant (TA), as a Graduate Assistant (GA), as well as a Research Assistant (RA) at UIC

Jan 2002–Jun 2002 I worked as a teacher in lectures given to high-school teachers under the program *Lifelong Learning: Familiarization with new technologies* which was supported by the Hellenic Ministry of Education due to the development project "Information Society"

Jan 2000–July 2000 I worked at Othisi[1] as a Computer Science teacher for the course *Developing Applications under a Programming Environment*

Oct 1998–Jul 1999 Head of the Mathematics department at the Students' Learning Support Center at Othisi. Othisi is a preparatory institute which prepares high-school students for entering Higher-Level Education (Universities, Technological Institutions)

Working Experience

Operating Systems. Linux, Mac OS, Solaris Unix, and all Microsoft operating systems.

Programming Languages. All major programming languages including, but not limited to, C, Objective C, C++, Visual Basic, Python, Cython, Pascal, LPA-Prolog, Haskell.

Miscellaneous. Model-View-Controller (MVC), Core Graphics, MapKit, NSURLConnection, P-threads, Message Passing Inteface (MPI), GNU Multiple Precision Arithmetic Library (GMP), Scalable Parallel Random Number Generators Library (SPRNG), Subversion, Git, Apache, SQLite, Oracle SQLPlus, Microsoft SQLServer, (X)HTML, PHP, CSS, TeX, LaTeX, XeTeX, GNUPlot, Maple, igraph, R, shell scripts in Unix / Linux / MS-DOS, VBScript, JavaScript.

[1] Mailing Address: Mitropoleos & Chaimanta 7, 151-25 Marousi, Attiki, Hellas, phone: +30-210-61.28.814, +30-210-61.43.812, fax: +30-210-80.61.353, homepage: `http://www.othisi.gr`

Theses

Master's Thesis. Real Solving on Algebraic Systems of Small Dimension. Master's Thesis, National and Kapodistrian University of Athens, Athens, Hellas, June, 2007.

Undergraduate Thesis. Application of Reinforcement Learning and Combinatorial Search to One-Player Games. Undergraduate Thesis, National and Kapodistrian University of Athens, Athens, Hellas, February, 2004.

Publications

7. Berger-Wolf, T., Diochnos, D.I., London, A., Pluhár, A., Sloan, R.H., Turán, G.: Commonsense knowledge bases and network analysis. In Commonsense, 2013.

6. Diochnos, D.I., Sloan, R.H., Turán, G.: On multiple-instance learning of halfspaces. In Information Processing Letters, 112(23): 933–936, 2012.

5. Diochnos, D.I.: Leveling-Up in Heroes of Might and Magic III. In Fifth International Conference on Fun with Algorithms (FUN 2010), Ischia Island, Italy, FUN 2010: 145–155, 2010.

4. Diochnos, D.I., Turán, G.: On Evolvability: The Swapping Algorithm, Product Distributions, and Covariance, In Fifth Symposium on Stochastic Algorithms, Foundations and Applications (SAGA 2009), Sapporo, Japan, SAGA 2009: 74–88, 2009.

3. Diochnos, D.I., Emiris, I.Z., Tsigaridas, E.P.: On the asymptotic and practical complexity of solving bivariate systems over the reals. In Journal of Symbolic Computation, 44(7): 818–835, 2009.

2. Διώχνος, Δ.: Επίλυση Αλγεβρικών Συστημάτων Μικρής Διάστασης στους Πραγματικούς. Στο Ετήσιο Βιβλίο με Επιλεγμένες Πτυχιακές και Διπλωματικές Εργασίες, Τμήμα Πληροφορικής και Τηλεπικοινωνιών, Εϑνικό και Καποδιστριακό Πανεπιστήμιο Αϑηνών, Ελλάδα, 5: 23–32, 2008.
(Diochnos, D.: Solving Algebraic Systems of Small Dimension over the Reals. In Annual Book of Selected Undergraduate and Graduate Theses, Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Hellas, 5: 23–32, 2008.)

1. Diochnos, D.I., Emiris, I.Z., Tsigaridas, E.P.: On the Complexity of Real Solving Bivariate Systems. In Proceedings Annual ACM International Symposium on Symbolic and Algebraic Computation (ISSAC), Waterloo, Canada, ISSAC 2007: 127–134, 2007.

Technical Reports

4. Diochnos, D.I.: Commonsense Reasoning and Large Network Analysis: A Computational Study of ConceptNet 4, arXiv:1304.5863 [cs.AI] .

3. Diochnos, D.I., Emiris, I.Z., Tsigaridas, E.P: On the complexity of real solving bivariate systems, INRIA RR 6116.

2. Diochnos, D.I., Emiris, I.Z., Tsigaridas, E.P.: Experimental implementation of more operations on algebraic numbers, possibly with the addition of numeric filters, and of robust operations on small polynomial systems, Algorithms for Complex Shapes with Certified Numerics and Topology, Workpackage I, Deliverable 1, Month 24, ACS-TR-241405-02.

1. Diochnos, D.I., Emiris, I.Z., Tsigaridas, E.P.: Benchmarks and evaluation of experimental algebraic kernels, Algorithms for Complex Shapes with Certified Numerics and Topology, Workpackage III, Deliverable 3, Month 24, ACS-TR-243306-02.

Software

SLV Maple Library. SLV is a library used in Maple. The acronym comes from *S*turm so*LV*er. It was developed as part of my master's thesis and solves univariate polynomials or bivariate polynomial systems using Sturm sequences. The solutions are (pairs of) Real Algebraic Numbers in Isolating Interval Representation.

Optimal Policy in Game Solo. An RL-agent that finds optimal policy in game Solo. The learning process is augmented through combinatorial search techniques.

Heroes of Might and Magic III. Different solvers for the general problem of Skill Advancing.

skills: Evaluation of user's policy based on skill trees and limited randomness, dimis, September 2009. Current version is 2.0 and supports five popular deterministic policies.

internals_mc: Evaluating Policies with Monte Carlo methods in Skill-Selection problem, dimis, July 2007. Current version is 2.0 and supports five popular deterministic policies with the use of the PTHREADS library.

ansa, ansaExtended: Solver for ANSA (AR) problem, dimis, April 2006. Source code for `ansa` is also available in GNU Multiprecision Arithmetic Library (GMP), although it is not necessary for practical applications. `ansaExtended` was developed in July 2006 in order to answer more interesting questions posed in Disjunctive Normal Form (DNF).

Inversion Distance and Sorting by Reversals. Tools that compute the inversion distance of two genomes as well as perform sorting by reversals between two genomes. Part of the source code was used as testbed in IOI-2004.

The Ellipsoid Method. The popular Ellipsoid Method used in Linear Programming, implemented in C.

Database for Undergraduate Courses. This is a program that can be used as a database for undergraduate courses passed at the Department of Informatics and Telecommunications as well as a tool for statistical analysis of the GPA and other departmental parameters which are crucial for graduate applications.

Talks

- Commonsense Knowledge Bases and Network Analysis, *Commonsense*, Ayia Napa, Cyprus, May 27, 2013.

- On Multiple-Instance Learning of Halfspaces. *X-Theory Day*, National and Kapodistrian University of Athens, Athens, Hellas, December 19, 2011.

- Evolvability in Learning Theory. Eötvös Loránd University, Budapest, Hungary, November 23, 2011.

- Evolvability in Learning Theory. University of Szeged, Szeged, Hungary, November 16, 2011.

- On Evolvability: The Swapping Algorithm, Product Distributions, and Covariance. *Algorithms Seminar*, National and Kapodistrian University of Athens, Athens, Hellas, December 23, 2010.

- Leveling-Up in Heroes of Might and Magic III. *Fifth International Conference on Fun with Algorithms (FUN 2010)*, Ischia Island, Italy, June 3, 2010.
- On Evolvability: The Swapping Algorithm, Product Distributions, and Covariance. *Eleventh International Symposium on Artificial Intelligence and Mathematics (ISAIM 2010)*, Fort Lauderdale, FL, USA, January 7, 2010.
- On Evolvability: The Swapping Algorithm, Product Distributions, and Covariance. *Midwest Theory Day, Fall 2009*, DePaul University, Chicago, IL, USA, December 5, 2009.
- On Evolvability: The Swapping Algorithm, Product Distributions, and Covariance. *Fifth Symposium on Stochastic Algorithms, Foundations and Applications (SAGA 2009)*, Hokkaido University, Sapporo, Japan, October 27, 2009.
- Implementation and Experiments on Real Solving of Bivariate Systems. *ACS Workshop*, Freie Universität, Berlin, Germany, May 9, 2007.