On the Complexity of Fundamental Computational Problems in Pedigree Analysis [1]

Antonio Piccolboni and Dan Gusfield

Computer Science Department
University of California, Davis.
Technical Report CSE - 99 - 8
September, 1999

[1]Research partially supported by grant DBI-9723346 from the National Science Foundation.

On the complexity of fundamental computational problems in pedigree analysis

Antonio Piccolboni and Dan Gusfield

September 28, 1999

Abstract

Pedigree analysis is a central component of many current efforts to locate genes that contribute to diseases or to valuable traits. The analysis usually involves solving one of two very computation-intense problems. We analyze the complexity of these two problems. Surprisingly, we show that both problems are NP-hard even for pedigrees that contain no inbreeding loops.

1 Introduction

"Rigorous analysis of human pedigree data is a vital concern in genetic epidemiology, human gene mapping, and genetic counseling" [1]. The key element of most of these analyses is the calculation or estimate of certain probabilities, or pedigree likelihoods. There are several methods and extensively used computer packages that compute these likelihoods for pedigrees of restricted forms [3, 4, 7]. However, no worst-case efficient method for computing the likelihoods in general pedigrees is known. "Evaluation of pedigree likelihoods remains a subject sorely in need of further theoretical improvement. Linkage calculations alone are among the most demanding computational tasks in modern biology" [1], often consuming months of computation in practice [2]. Linkage and pedigree analysis is almost always the key first step in positional cloning, a very successful approach to finding the location of genes contributing to certain diseases, or to favorable traits of economic importance in agriculture. There is an enormous literature on biological studies where linkage and pedigree analysis is a critical element.

At the heart of most pedigree/linkage analyses is the computation of the probability (and hence likelihood) of the observed data, given that some data in the pedigree is missing. Alternatively, the computation must find values for the missing data to maximize the probability of the entire data. In linkage analysis, in addition to missing data there is also the complication of genetic recombination, so that the probability computations must be done repeatedly at the inner loop of the linkage analysis.

Surprisingly, given the extreme importance of the calculations involved in pedigree and linkage analysis, the large literature on computing and using them, and the long computation times encountered in practice, the basic question of whether their calculation is an NP-hard problem has not been addressed in the literature. In this paper we establish that even a small deviation from the special cases where efficient computation is known to be possible, leads to an NP-hard problem.

This basic result can be extended to establish NP-hardness of certain approximations, and to a variety of different questions concerning pedigree analysis. We also note, however, that certain efficient approximations are possible. These additional results will be detailed in a more complete version of this paper.

2 Introduction to pedigrees and genetic models

Definition 1 *A* pedigree *is a directed acyclic graph $G = (V, E)$ where the indegree of every vertex is either 0 or 2 and whose corresponding* marriage graph *(to be defined shortly) is bipartite.*

Definition 2 *Given a directed graph $G = (V, E)$, the marriage graph of G is $M = (V, E')$, where $E' = \{(v, w) : v, w \in V$ and there exist z s.t. $(v, z), (w, z) \in E\}$.*

The intended meaning of a pedigree graph is that a node in a pedigree represents an individual in a population, and an arc from node v to node w means that individual v is a parent of w. The indegree constraint reflects the standard convention in pedigree analysis that either both parents of an individual or none belong to a pedigree. In the later case, the individual is called a *founder*. The pedigree graph must be acyclic because no individual can be an ancestor of himself. In the context of a pedigree, when two individuals "mate" or are "mates", they share an offspring in the pedigree.

The marriage graph is the undirected graph where the nodes represent the same individuals as in the pedigree, but there is an undirected edge between two individuals if and only if they mate. Clearly, a marriage graph must be bipartite to comply with gender distinction. Given a digraph, it's easy to check whether it is a valid pedigree, testing the defining properties with well known algorithms.

Even though a pedigree is acyclic, the associated undirected graph may have cycles, that are called *loops* in the pedigree literature. Two kinds of loops are distinguished: when two individuals sharing a common ancestor mate, the loop is an *inbreeding* loop, otherwise it is a *marriage* loop. Marriage loops are natural in real pedigrees, occurring whenever two siblings and both parents are in the pedigree. Inbreeding in pedigrees is more common in some species than in others. It is relatively uncommon, but definitely a possibility, in human pedigrees, while very common in many domestically bred animals.

The occurrence of inbreeding loops is commonly perceived by practitioners as the main source of computational difficulty in pedigree analysis, even

though the only known polynomial-time algorithms for those computations assume pedigrees containing no marriage or inbreeding loops. In this paper, we provide an explanation for this, countering the common perception. We show that the central computational tasks in pedigree analysis are NP-hard even when there are no inbreeding loops, as long as marriage loops are permitted.

In addition to the graph (pedigree), a full problem instance associates two types of random variables to each node. These types are *genotypic* and/or *phenotypic*. For some of the nodes, the value of these random variables are given as input, and are assumed correct, and for others the values are completely unknown. The fundamental problems in pedigree analysis arise from these unknown or missing data.

A genotypic random variable is a vector of pairs of states called *alleles* and is a model for the information carried by the DNA. The positions of the vector are called *loci*. A phenotypic random variable is related to an observable feature of an organism and is the result of the interaction between the genotype and the environment. The separation between phenotype and genotype is important when the genotype must be inferred from the phenotype (which has typically been the case until recently).

Finally, we need to specify the model for the joint distribution of these random variables. In pedigree analysis, it is often assumed that the phenotype of an individual casually depends only on its genotype, which in its turn depends only on the genotype of its parents. These casual independence assumptions result in a precise statement about the joint probability distribution of all the random variables associated with a pedigree. Let $P(g_i)$ be the probability distribution of the genotype of founder i, let $P(g_i|g_{f(i)}, g_{m(i)})$ be the conditional probability of the genotype of non-founder i given the genotype of its father and mother ($f(i)$ and $g(i)$ respectively), and finally let $P(y_i|g_i)$ be the probability of the phenotype of individual i given its genotype. In genetics jargon, these are known as *prior, transmission* and *penetrance* probability respectively.

Then the joint probability distribution of all the genotypes, G, and phenotypes Y is just the product of the three above terms, that is

$$P(G,Y) = \prod_{\text{founder } i} P(g_i)P(y_i|g_i) \prod_{\text{non founder } i} P(g_i|g_{f(i)}, g_{m(i)})P(y_i|g_i)$$

The pedigree, along with the prior, transmission and penetrance probabilities form the input to an instance of a pedigree analysis problem. However, for the input to represent a meaningful genetic model, it is necessary to specify further constraints on the probability distributions. The prior distribution can be any discrete distribution. But the transmission distribution must reflect genetic reality, which is most often described by simple Mendelian genetics (parts of which are detailed later in the paper).

The model specified above is a special case of a family of statistical models known as Bayesian networks. In the general case of Bayesian networks, general directed, acyclic graphs and unrestricted probability distributions are allowed. It is known in the Bayesian network literature that computing certain probabilities over such models is NP-hard [6, 5]. However, these proofs use graphs and

transmission models that are totally unrealistic as a genetic model, so they do not imply the hardness of the equivalent problems when restricted to pedigrees and genetically sound transmission models.

3 The fundamental computational problems in pedigree analysis

The following two problems are at the heart of most problems in pedigree analysis, and their solution is often in the inner loop of pedigree analysis algorithms.

Problem 3 Marginal-probability: *Given a pedigree and, for every node, its prior distribution or transmission distribution (whichever appropriate) and penetrance distribution and given values for a subset G' of the genotype G and a subset Y' of the phenotype Y, compute* $P(G', Y') = \sum_{G \setminus G', Y \setminus Y'} P(G, Y)$.

This amounts to taking into account all possible explanations of the data, that is all values of G and Y compatible with G' and Y'. Of course, this is only an issue when either G' is a strict subset of G, or Y' is a strict subset of Y. Sometimes it is necessary to single out only one such explanation, the one with the highest likelihood. This leads to a second

Problem 4 Maximum-likelihood: *Given a pedigree and, for every node, its prior distribution or transmission distribution (whichever appropriate) and penetrance distribution and given values for a subset G' of the genotype G and a subset Y' of the phenotype Y, compute* $\max_{G \setminus G', Y \setminus Y'} P(G, Y)$.

4 NP-hardness results for the marginal and maximum-likelihood problems on pedigrees

We assess the complexity status of the two problems above by restricting to special cases. The NP-hardness of these special cases implies the NP-hardness of the general problems. We study a single locus, where each of the two "copies" can take on at most two possible states (alleles). Furthermore, we assume that the trait of interest is *codominant* and *fully penetrant*. What this means is that the phenotype is in a deterministic, one to one relation with the genotype, and hence each of the three combinations of alleles specifies a different phenotype. Therefore, we will remove any reference to phenotype and deal directly with the genotype. We consider only a single random variable for each individual, which can take on one of three values denoted 00, 11, and 01, for concreteness. As transmission distribution of alleles between parents and offspring, we assume standard Mendelian probabilities without mutation. That is, if the parent's genotype is 00 or 11, it passes on 0 or 1, respectively, with certainty; and if the parent is 01, it passes on 0 or 1, respectively, with probability 1/2. Finally, we assume a uniform distribution of the genotypes of the founders, meaning that

$P(g_i = 00) = P(g_i = 01) = P(g_i = 11) = 1/3$, and all founders are independent. In the construction, we will not use any inbreeding loops.

The particular restrictions we assume do not limit the relevance of the result, but rather expand it. The NP-hardness result is established with a model that reflects the most basic, and simplest biological case. Any other realistic biological model is almost certain to contain this one as a special case, hence the results proved here establish that marginal and maximum-likelihood problems are NP-hard in those models as well.

4.1 The maximum-likelihood problem

In the next sections we prove the following theorem:

Theorem 5 *The maximum-likelihood problem is NP-hard.*

4.1.1 The reduction

We define a reduction from 3SAT with at most three occurrences per variable, that is for every instance Σ of problem 3SAT we define an instance of the maximum-likelihood problem, MaxL(Σ), such that it is satisfiable if and only if the probability of a solution for MaxL(Σ) is larger than a given threshold.

We can assume that every variable appears both positive and negated (otherwise it can be easily eliminated together with all the clauses containing it). From the 3SAT instance, we construct a pedigree in polynomial time, as follows. For every variable x in Σ, the set of founders includes four individuals, x, x', $\bar{x}$ and $\bar{x}'$. In order to ensure that the marriage graph is bipartite, we arbitrarily assume that individuals corresponding to negated variables are male whereas the ones corresponding to positive variables are female. Individual x mates $\bar{x}$ and $\bar{x}'$ $k+1$ times each, where $k = 5$ (but we will need larger k for other proofs later on, so we describe the construction for any k). Among the offspring, k have genotype 11, and one has genotype 10. Similarly, individual x' mates $\bar{x}$ and $\bar{x}'$ $k+1$ times each, with the same distribution of offspring genotypes (Figure 1). We call the sub-pedigree just described the "variable gadget".

In addition to variable gadgets, we construct one "clause gadget" for each clause in Σ. The construction of these gadgets depends on whether the clause contains two or three literals and on whether all literals are either all positive or all negated or mixed. For each clause with two literals l_1, l_2, one positive and one negated, the individuals denoted by l_1, l_2 and those denoted by l_1', l_2' mate once and their offspring mate once giving birth to an individual whose genotype is 00 (Figure 2).

For a clause C with three literals l_1, l_2, l_3 we need to distinguish two cases. If the literals are not all positive or all negated, let us assume they are ordered so that l_1 and l_2 are not both positive or negated and the same is true of l_2 and l_3, l_1' and l_2', l_2' and l_3'. Then, in the clause gadget for that clause, l_1 mates l_2 and l_1' mates l_2'. Their offspring, C_1 and C_2, mate l_3 and l_3' respectively, issuing two more offspring, C_3 and C_4. These mate in turn and their offspring, C_5, has genotype 00 (Figure 3).

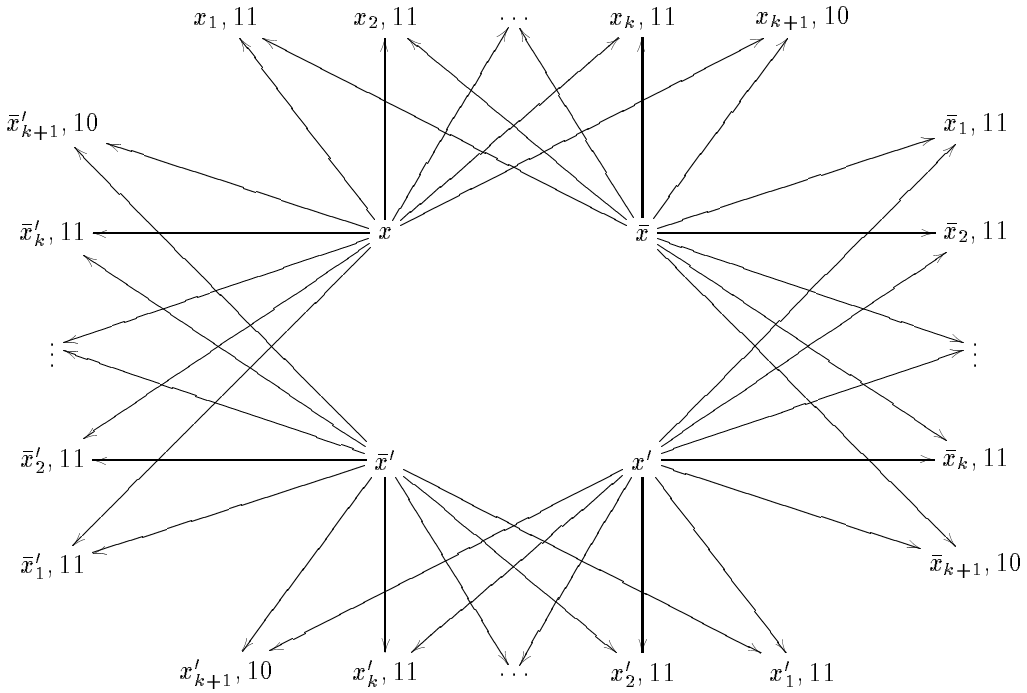

Figure 1: The variable gadget for variable x

If a clause has only positive or only negated literals, then the above constructions would mate two individuals corresponding to positive or negative literals. Under our interpretation of positive literals as females and negated literals as males, this would mate two individuals of the same gender and hence make it less clear whether the marriage graph of MaxL(Σ) is bipartite. We can fix this with the following simple modification. We introduce a new founder with genotype 11, with the opposite gender of l_1 and we mate it with l_1. We declare the offspring to have the opposite gender of l_1. We then use this offspring in place of l_1 in the above construction.

Let c be the number of clauses and v the number of variables in MaxL(Σ). The correctness of the reduction rests on the following:

Lemma 6 *Σ is satisfiable iff the optimum for MaxL(Σ) is larger than*

$$\frac{1}{3^{4v}}\frac{1}{2^{4(k+1)v+3c}}$$

4.1.2 The proof

The proof is implied by two lemmas proved in this section.

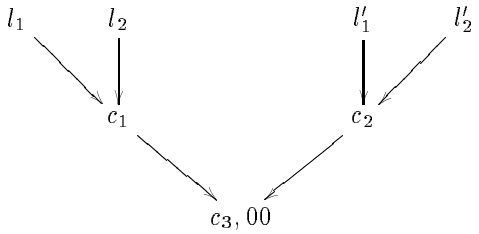

Figure 2: The clause gadget for clause $C = l_1 l_2$

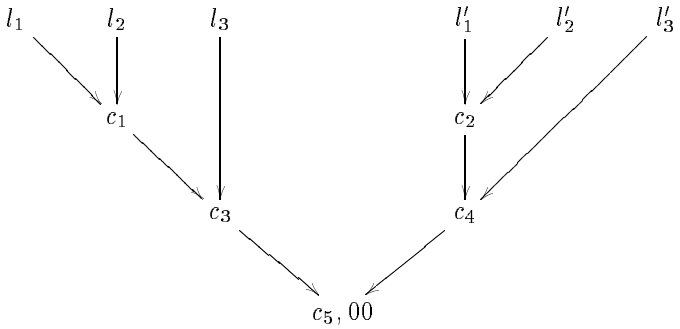

Figure 3: The clause gadget for clause $C = l_1 l_2 l_3$

Lemma 7 *If there exists a satisfying assignment for Σ then there is a setting of the genotypes for the pedigree MaxL(Σ) whose probability is greater or equal than the bound in Lemma 6.*

Proof: If x is true in the satisfying assignment then set individuals x and x' to 10, and set $\bar{x}$ and $\bar{x}'$ to 11. If x is false, then set x and x' to 11, and set $\bar{x}$ and $\bar{x}'$ to 10. Then set every unknown non-founder genotype to 10, unless both parents have genotype 11; in that case, obviously, set it to 11. Simple calculations show that these genotypes have the required probability. ■

This establishes one direction of Lemma 6. In order to establish the other direction, we need to first introduce some additional concepts and terminology.

First observe, from the variable gadget, that none of the founders can be a 00 (or, equivalently, such a genotype occurs with probability 0), nor can a variable and its negation both be 11. It follows that in a single variable gadget, the are only three ways to set the genotypes of the four individuals to yield a non-zero probability: either set all the individuals to 10, or set three to 10 and one to 11, or set two to 10 and two to 11. Moreover, in the later case, it must be that both x and x' are 10, or that both are 11. We call the first two settings of a single variable gadget *inconsistent* and the third setting *consistent*. Any setting of the genotypes in the entire pedigree is said to be consistent if and

only if the setting of each variable gadget is consistent.

A choice of genotypes for all individuals is *satisfying* if and only if for each clause C, in the associated clause gadget at least two founders have genotype 10, one on the maternal side of individual C_3 for two-literal clauses or C_5 for three-literal clauses (see Figures 2 and 3), and one on the paternal side. We observe that consistent satisfying genotypes of MaxL(Σ) are in a one-to-one correspondence with satisfying assignments of Σ. Moreover, a non-satisfying genotype has zero probability because either all the maternal or all the paternal ancestors of C_3 (C_5 for three-literal clauses) would be 11, making it impossible for C_3 to be 00.

Lemma 8 *If there is a setting of the genotypes for pedigree MaxL(Σ) whose probability is greater or equal to the bound in Lemma 6, then there exists a satisfying assignment for Σ.*

Proof:

All non-satisfying settings of the genotype yield a probability of zero, so the assumed setting of genotypes for MaxL(Σ) must be a satisfying setting. As observed, any consistent, satisfying setting of the genotypes specifies a way to set the variables to satisfy Σ. Thus it is enough to show that no inconsistent setting of the genotypes can satisfy the probability bound given in Lemma 6.

Suppose h of the variable gadgets are set inconsistently. Then the transmission probability of one of these gadgets is at most $\frac{1}{2^{6k+4}}$. The transmission probability of each consistent variable gadgets is $\frac{1}{2^{4k+4}}$. There are at most $3h$ clauses containing variables whose gadgets where set inconsistently. The transmission probability of these clause gadgets is at most 1. For other clauses, the transmission probability is at most 1/8. The founders probability is the same as above. Putting the product together:

$$P(G) \leq \frac{1}{3^{4v}} \frac{1}{2^{(6k+4)h}} \frac{1}{2^{(4k+4)(v-h)}} \frac{1}{2^{3(c-3h)}} \tag{1}$$

This is less than the required bound whenever $k \geq 5$. ■

We have therefore established both sides of Lemma 6 and have shown that the maximum-likelihood problem in NP-hard.

4.2 The marginal-probability problem

Theorem 9 *The marginal-probability Problem is NP-hard*

The reduction is similar, but with a different choice of k. The key observation is that if the given formula of Σ is not satisfiable, then every satisfying genotype must be inconsistent, and we can make the probability of inconsistent assignments very small by increasing k. Conversely, if Σ is satisfiable than there is a consistent satisfying genotype with a relatively large probability. The only technical detail is to choose k large enough so that the sum of the probabilities of all inconsistent genotypes is less than the probability of one satisfying,

consistent genotype. As stated above, the probability of a satisfying, consistent genotype is larger than

$$\frac{1}{3^{4v}}\frac{1}{2^{4(k+1)v+3c}} \tag{2}$$

There are at most 2^{6c+4v} inconsistent non-zero-probability genotypes, whose probability is at most (see Equation 1)

$$\frac{1}{3^{4v}}\frac{1}{2^{(6k+4)h}}\frac{1}{2^{(4k+4)(v-h)}}\frac{1}{2^{3(c-3h)}}$$

where h is, again, the number of variables whose variable gadget is set to be inconsistent. The product of the latter two quantities is smaller than (2) for $k > 9/2 + \frac{3c+2v}{h}$ and, since there is at least one "inconsistent" variable in every inconsistent genotype, this is true for $k > 9/2 + 3c + 2v$.

5 Discussion

We have established that the fundamental computational problems in pedigree analysis are NP-hard. Moreover, this is true even in pedigrees that do not contain inbreeding loops. While hardness results are standard in the computer-science, such results are quite new in the field of pedigree analysis and computational genetics. As these areas grow in importance, and of interest to an expanding group of computational scientists, we believe it is very helpful to map out what central problems are likely to lack efficient (worst-case) deterministic solutions. Knowing that the problem is NP-hard should not lead to its abandonment, but should focus or justify alternative efforts to obtain practical solutions.

To some, it is "intuitive" (and therefore, not in need of rigorous analysis) that the problems discussed here could not have an efficient (worst-case, deterministic) solution, because there are an exponential number of terms, as a function of pedigree size, in the definition of the likelihood. That is, there are an exponential number of legal ways that the incomplete data can be specified. More simply, there is an exponential number of paths describing the way a gene can flow to an individual from one of its ancestors. However, the argument that efficient computation is not possible when the solution-space grows exponentially with the pedigree size, cannot be relied on, as there are counter-examples to that "intuition" in pedigree analysis (and combinatorics generally).

For example, when a pedigree has inbreeding loops, it is possible for an individual to obtain both copies of some gene from one single ancestor in the pedigree. When this happens, the allele at that locus is said to be *identical by descent*. The probability, under the Mendelian transmission model, that a given allele is identical by descent for an individual is called the *inbreeding coefficient* for that individual. Computing the inbreeding coefficients is a basic task needed in some pedigree analyses. Methods for calculating the inbreeding coefficient described in the biological literature typically follow the definition and involve enumerating all node-disjoint pairs of paths from an ancestor to an individual.

That set of paths can grow exponentially with the size of the pedigree, and lead to the intuition that the inbreeding coefficient cannot be computed efficiently. However, the inbreeding coefficient can be computed in worst-case polynomial time via dynamic programming recurrences [8, 1, 9]. Hence, intuition alone cannot be relied on, and it is worth rigorously establishing which pedigree computation problems can be solved efficiently and which are NP-hard.

6 Acknowledgements

This research has been partially supported by grant DBI-9723346 from the National Science Foundation.

References

[1] K. Lange, "Mathematical and Statistical Methods for Genetic Analysis", Springer, 1997.

[2] S. Lin, Monte Carlo methods in genetic analysis, *in* "Genetic Mapping and DNA Sequencing," (T. Speed and M.S. Waterman, Eds.), pp. 15-38, Springer, Berlin, 1996.

[3] R.C. Elston and J. Stewart, A general model for the analysis of pedigree data, Human Heredity, 21:523-542, 1971

[4] E.S. Lander and P. Green, Construction of multilocus genetic linkage maps in humans, Proceedings of the Natl. Acad. Sci. USA, 84:2363-2367

[5] P. Dagum and M. Luby, Approximating Probabilistic Inference in Bayesian Belief Networks is NP-Hard, Artificial Intelligence 60(1):141-153, 1993

[6] G. Cooper, The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks Artificial Intelligence 42:393-405, 1990

[7] R.W. Cottingham Jr. and R.M. Idury and A.A. Schäffer, Faster sequential genetic linkage computations, Am. J. Human Genetics 1993, 53:252-263

[8] E. Thompson, "Pedigree Analysis in Human Genetics", The Johns Hopkins University Press, 1986.

[9] A.A. Schäffer, Computing Probabilities of Homozygosity by Descent. Genetic Epidemiology, 16:135–149, 1999.