# The computational hardness of feature selection in strict-pure synthetic genetic datasets

by

©Majid Beheshti Mohtasham

A Dissertation submitted to the School of Graduate Studies in partial fulfillment of the requirements for the degree of

**M.Sc.**

**Department of Computer Science**

Memorial University of Newfoundland

**September 2019**

St. John's Newfoundland

# Abstract

A common task in knowledge discovery is finding a few features correlated with an outcome in a sea of mostly irrelevant data. This task is particularly formidable in genetic datasets containing thousands to millions of Single Nucleotide Polymorphisms (SNPs) for each individual; the goal here is to find a small subset of SNPs correlated with whether an individual is sick or healthy(labeled data). Although determining a correlation between any given SNP (genotype) and a disease label (phenotype) is relatively straightforward, detecting subsets of SNPs such that the correlation is only apparent when the whole subset is considered seems to be much harder. In this thesis, we study the computational hardness of this problem, in particular for a widely used method of generating synthetic SNP datasets.

More specifically, we consider the feature selection problem in datasets generated by "pure and strict" models, such as ones produced by the popular GAMETES software. In these datasets, there is a high correlation between a predefined target set of features (SNPs) and a label; however, any subset of the target set appears uncorrelated with the outcome.

Our main result is a (linear-time, parameter-preserving) reduction from the well-known Learning Parity with Noise (LPN) problem to feature selection in such pure and strict datasets. This gives us a host of consequences for the complexity of feature selection in this setting. First, not only it is NP-hard (to even approximate), it

is computationally hard on average under a standard cryptographic assumption on hardness on learning parity with noise; moreover, in general it is as hard for the uniform distribution as for arbitrary distributions, and as hard for random noise as for adversarial noise. For the worst case complexity, we get a tighter parameterized lower bound: even in the non-noisy case, finding a parity of Hamming weight at most k is W[1]-hard when the number of samples is relatively small (logarithmic in the number of features).

Finally, most relevant to the development of feature selection heuristics, by the unconditional hardness of LPN in Kearns' statistical query model, no heuristic that only computes statistics about the samples rather than considering samples themselves, can successfully perform feature selection in such pure and strict datasets. This eliminates a large class of common approaches to feature selection.

# Acknowledgements

Fist, I would like to express my sincere gratitude to my M.Sc. supervisor Prof. Saeed Samet for the continuous support of my master's study and research, for his patience, motivation, enthusiasm, and great knowledge. His guidance helped me in all the time of my graduate studies. I could not have imagined having a better supervisor for my master's study.

Also, I must express my special thanks to the second supervisor Prof. Antonina Kolokolova who always cared about my situation during the research time. Her integrity, motivation, deep knowledge in complexity theory, cooperation and sound understandings about student's circumstances which played a key role in this thesis. This thesis could not be completed without her.

Besides my supervisors, I would like to thank the rest of my thesis committee: Prof. Manrique Mata-Montero and Prof. Hamid Yousefi for their review. Also, Special thanks to Prof. Manrique Mata-Montero for his insightful comments.

Last but not the least, I would like to thank my family for supporting me spiritually throughout my life, also my wife Zohreh for her patience and integrity during completing this thesis.

# Table of Contents

# List of Tables

# List of Figures

# List of Abbreviations

| | |
|---|---|
| DNA | Deoxyribonucleic Acid |
| DNF | Disjunctive Normal Form |
| GA | Genetic Algorithm |
| GAMETES | Genetic Architecture Model Emulator for Testing and Evaluating Software |
| LPN | Learning Parity with Noise |
| MAF | Minor Allele Frequency |
| MDR | Multifactor-Dimensionality Reduction |
| PAC | Probably Approximately Correct |
| SNP | Single Nucleotide Polymorphism |
| SQ | Statistical Query |
| SVM | Support Vector Machines |
| T2D | Type 2 Diabetes |

# Chapter 1

# Introduction

Genetic datasets which are produced in health care system play an important role in human lives. These datasets with huge amounts of people's information are employed in detecting genetic reasons of various diseases. One of the main parts of research areas which work on genetic data is the genetic analysis of complex traits. Complex traits result from gene-gene and gene-environment interaction [HY08]. Complex traits show the important function of epistasis or gene-gene interaction in genetics research [CW97,SAK05]. The study of genetic data has gradually shifted to genome-wide association studies, which are the case–control studies for considering single nucleotide polymorphisms (SNPs) in detecting genetic factors associated with complex diseases. In such studies of SNP data of complex diseases, the problem arises when each individual SNP has no significant effect on the phenotype/disease, however the combination of SNPs has a strong effect [LGW+14]. This specific aspect makes the key task of feature selection in the SNP datasets more difficult.

There are many methods and algorithms developed for feature selection in SNP datasets [Cor09,MS16,YLCC13,İT13b,SI07,YWCY17,UGEFC12,CLCY12], with the goal to help detect relevant SNPs associated with disease. Due to the high compu-

tational complexity of solving this specific feature selection, heuristic methods are employed. Based on the nature of heuristics, they are not guaranteed to provide a solution for the problem.

The starting point of this thesis is understanding the complexity of feature selection in SNP datasets with the focus on complex traits. Specifically, we analysed the complexity of feature selection in pure and strict datasets generated by popular software which is called *GAMETES* [UKSA+12]. The pure-strict data models, which are produced by the GAMETES, include the hardest case of SNP epistasis in terms of detecting the associated SNPs with disease [UKSA+12].

In this thesis, regarding studying the computational complexity of feature selection in the pure and strict GAMETES datasets (we call it GAMETES problem), a known problem in learning theory area is considered. Learning parity with noise (LPN) is a hard problem under the standard cryptographic assumption [Pie12] and has a number of conditional and unconditional lower bounds. In this problem, given a list of labeled binary strings (samples), where the label is the parity of a specific subset of bits (however, the label may be wrong with small probability), the goal is to find the bits parity of which determines the label [BKW03, KMV08, Reg09, Pie12, KMV08]. Here, we show that the GAMETES problem is as hard as LPN by reducing LPN to the GAMETES problem. This lets us get unconditional hardness results for the statistical query model (and some models of differential privacy), as well as parameterized complexity worst-case lower bounds for the GAMETES problem. We also present experimental evidence that even for the state-of-the-art feature selection heuristic ReliefF accuracy on GAMETES datasets decreases dramatically with the increase in the number of features.

This thesis is organized as follows. Chapter 1 introduces the general aspects of the study. In chapter 2, the completed previous work of the feature selection studies is

shown. Moreover, the feature selection in genetic datasets is specifically represented based on heuristic methods and the procedure of generating GAMETES models is presented in this chapter. Chapter 3, demonstrates the background of learning theory and complexity which is needed for better understanding the results. The main results of this thesis is provided in chapter 4 including defining GAMETES problem in theory mode, and reductions from LPN to GAMETES problem. Finally, conclusions and future work are covered in chapter 5.

# Chapter 2

# Feature selection in genetic datasets

In this chapter, we survey feature selection in genetic datasets, particularly in Single Nucleotide Polymorphism(SNP) datasets. First, genetic datasets and their aspects are introduced. Then the consequences of employing prevailing feature selection methods on genetic datasets are discussed. Accordingly, heuristic feature selection methods, which are related to our thesis, are considered and analyzed in general. Finally, in this chapter, the specific synthetic SNP data, which served our study, is introduced.

The genetic data that are studied in this work include the health and non-health-related information of individuals. The data are prepared and processed for study or clinical purposes [SB18]. Improvements in bioinformatics and genetics result in generating huge amounts of genetic data in clinical and research environments [SB18]. In order to learn genetic data and detect potential interactions between diseases and hidden genetic factors, studying the genetic data is essential [SB18,Kno14,CH13]. For example, employing data mining and machine learning methods is extremely valuable for classification, prediction, diagnosis, and other purposes. Despite this value, data

scientists face high dimensional data which cause many problems. These problems, which arise from irrelevant features of datasets, include high computational complexity that results in wasting time and resources. Thus, feature selection is highly essential in machine learning and data mining of genetic data.

In this thesis, we worked on SNP datasets that have their specific characteristics such as interactions between SNPs. One of the most important tasks in studying SNPs is detecting SNP interactions that are correlated with common diseases. This performs an important and enhancing duty in explaining the genetic foundation of disease susceptibility as well as provides support to invent new diagnostic tests and treatments [LGW+14]. To remove irrelevant SNPs (features), which are plentiful, and identify relevant features (SNPs which are associated with disease together), feature selection in SNP datasets becomes crucial. Thus, feature selection is inevitably needed in SNP datasets for doing any data mining process.*

## 2.1 Previous work

Feature selection is an important tool for data scientists since it can help to extract hidden knowledge from an enormous amount of data particularly in health and genetic datasets. A lot of work have focused on improving the performance of data mining in this area. The previous related work on genetic data feature selection can be categorized into two general groups: first, studies which mention and evaluate various feature selection methods, however main focus is working on classifications to predict and diagnose a disease, and second, studies which mainly concentrate only on different feature selection methods used for genetic datasets of a particular disease.

With regard to the first part, studies which evaluate various classification methods

*a comprehensive interpretation of SNP datasets will be presented in 'Synthetic datasets' section.

to predict and diagnosis of a disease, the elements of the interaction of Multi-SNPs is found in [MM09] by discussing the potential of applying the Apriori-Gen algorithm to the association study for the type 2 diabetes. General reviews are done in [KTS+17] to show the systematic efforts of identifying and reviewing machine learning and data mining approaches which are applied on diabetes mellitus work. This global review addresses a wide range of related methods and techniques of data mining and machine learning. Karegowda, V, Jayaram, and Manjunath [KPJM12] used K-means clustering to identify and prevent incorrectly classified instances. The correctly classified instance by K-means is used as input to decision tree after conversion of continuous data to categorical data. The proposed cascaded shows improved classification for PIMA diabetic dataset. In [PSGK16] the authors used a decision tree method and proposed models with higher performance to classify diabetic patients, across three age groups in the Canadian population. Zheng et al. [ZXX+17] proposed an accurate and well-organized framework to identify subjects with and without type 2 diabetes mellitus from electronic health records. This study forces machine learning to automatically extract patterns of type two diabetes mellitus. In addition, they boost the predictive power by overcoming the extensive separation rage of cases and controls in professional algorithms. Another study is done in this field about using machine learning approaches in prediction and diagnosis of diabetes mellitus [VA15b]; they proposed decision support system for prediction which utilizes decision stump as the base classifier in the AdaBoost algorithm. Barakat, Bradley, and Nabil. H [BBB10] improved a hybrid method for medical diagnosis. Surprisingly, they employed Support Vector Machines (SVMs) for the diagnosis and prediction of diabetes, where an extra rule-based explanation element is employed to provide comprehensibility. Vijayan and Anjali [VA15a] showed that decision support systems are helping specialists in analyzing different patterns of a disease. They also proposed a computerized information

system to predict diabetes after conducting a detailed study of techniques like individual classification, AdaBoost and Stacking. Cheruku, Edla, and Kuppili [CEK17] also used a spider monkey-based rule miner for classification of diabetes datasets; the proposed algorithm reached to the best ranking in average sensitivity and the second-best ranking in average classification accuracy in comparison with several standard meta-heuristic-based rule mining algorithms. SVMs are also employed as a beneficial tool for classification in bioinformatics [BHOP10, SCY$^+$16, KC13]. Shen et al. [SCY$^+$16] employed SVM for classification of some well-known medical datasets. A machine learning classification approach is proposed in studying differences in hand movement and muscle coordination between healthy subjects and Parkinson's disease patients in [KKHVA17]; as the authors mentioned, the feature selection process is essential in this article. In the statistical genetic filed, a classification of two-locus models with continuous penetrance values is done in [HY08]. The authors provided a complete classification of biallelic two-locus models. In addition, they solved the classification problem for dichotomous trait disease models and provided a complete framework for studying epistasis in Quantitative Trait Locus (QTL) data. QTL is a genomic region (section of DNA) responsible for the variation of a quantitative trait [MRRR$^+$16]. The authors [HY08] discussed the connection between their classification and standard epistatic models.

Regarding to the second part which is related to studies which principally focus on the feature selection methods on particular disease datasets, Wang et al. [WWC16] surveyed feature selection methods for big data in bioinformatics. The authors described the common categorizations for feature selection methods, then formulated these methods in a new categorization based on a search problem viewpoint. Exhaustive search, heuristic search, and hybrid methods are three new classes for the feature selection in big data based on this new point of view; sub-categories are introduced for

some of the main categories. The authors referred to many different methods which worked on genetic datasets with their advantages and disadvantages. In another work, a hybrid algorithm used to select tag SNPs in [İT13a]. These tag SNPs are the best selected SNPs or features which increase the prediction accuracy. By their results, they proved that the proposed method has higher performance in selecting tag SNPs than other mentioned methods in the article. Hira and Gillies [HG15] provided a notable review of performing the dimensionality reduction on high-dimensional micro array data with biological platform. The authors focused on summarizing different feature selection methods with their advantages and disadvantages of each method to save the computational time and resources for one who decides to employ the method. In [GPDHGPR13], the researchers proposed an evolutionary simultaneous instance and feature selection algorithm, that is scalable to high amounts of instances and features. This algorithm is based on the divide-and-conquer principle combined with bookkeeping. Also, this study claimed that the implementation of the proposed algorithm is easy and can be used in a parallel environment. In another work [LPH$^+$13], Liu et al. investigated four feature selection methods on different disease datasets; in this study, t-test, significance analysis of microarrays (SAM), rank products (RP), and random forest (RF) are tested on four different disease datasets. Each disease contains three cross-lab datasets and the authors ranked these mentioned methods based on their performance at the end of the study. Hybrid methods and evolutionary based methods are also used for feature selection in genetic datasets. For instance, Boutorh and Guessoum [BG16] proposed a hybrid method including Association Rule Mining (ARM), Neural Network (NN), Grammatical Evolution (GE), and Genetic Algorithm (GA) to remove irrelevant features in SNP datasets. They applied the proposed method on complex disease SNP dataset and compared it with different combined methods which failed versus the high performance of the pro-

posed method. In [MG13], Maji and Garai presented a feature selection method based on fuzzy-rough sets by maximizing both relevance and significance of the selected features. In this work, the proposed method is compared to different existing feature selection methods on different genetic datasets; this method showed better relevant features with lower cost. Ban et al. [BHOP10] gained benefit from SVM-based feature selection method to identify type 2 diabetes associated SNPs. Mostofi and Sadikoglu [MS16] also compared different evolutionary algorithms in detecting the associated SNPs (detecting relevant features) to breast cancer; this paper evaluated various methods with different configurations and demonstrated that Gauss particle swarm optimization (GPSO) achieved higher accuracy compare to other evolutionary method. The article of Li et al. [LGW$^+$14], showed an overview of SNP study in recent years in genome-wide association study; the authors discussed principles and efficiency and compared different methods in this area. In this study [LGW$^+$14], it is concluded that new computational methods based on attribute selection address the regulation of computational cost and effects to increase SNP interaction detection. In another work [Cor09], the author provided a survey of machine learning methods which is used to detect interactions between genetic loci that correlated with human genetic disease. The computational time and implementation analysis, of which several popular machine learning methods to find the associated SNPs to a particular disease, is discussed in this review. Yang et al. [YLCC13] also remarked Improved Genetic Algorithms (IGA) to generate genotype SNP barcodes for assessing of breast cancer susceptibility. The authors further pointed out that the results proved the ability of the IGA in identifying the best fitness of cases and controls; this method, in the condition of huge number of SNPs, can possibly be employed to detect complex gene-gene SNP interactions involved in genome-wide association studies. In [İT13b], the authors proposed a method to select the tag SNPs and predict the rest of SNPs in

genes; additionally, the method advocated in the study is referred to as CLONTagger method with parameter optimization, which uses the SVM and Clonal Selection Algorithm (CLONALG) to predict the rest of SNPs and select tag SNPs, respectively. This study used Particle Swarm Optimization (PSO) to optimize SVM parameters and it is concluded that this method, based on the experimental results, reached to higher accuracy than other tested methods mentioned in this article. In the same paper [SI07] in terms of identification of associated SNPs, a logic regression method (based on a combination of bootstrap and logic regression) is proposed to identify SNP interactions explanatory for the disease status in a case–control study; this method is applied on both simulated and real SNP datasets. In another study [YWCY17], Yang et al. introduced a hybrid method called Dynamic Center Particle Swarm Optimization K-Nearest Neighbors (DCPSO-KNN), to detect SNP-SNP interactions which are related to chronic dialysis. Their experimental results indicated that the proposed method improved searching efficiency for SNP-SNP interactions associated with the potential risk of chronic dialysis. Yang et al. [YMLC16] addressed a type of Genetic Algorithm (GA), based on local search method, to detect significant genetic association models between large numbers of SNP combinations. This algorithm called lsGA which employed two disease models to simulate the large data sets considering the minor allele frequency (MAF), number of SNPs, and number of samples; the study showed that the proposed method can detect more relevant SNPs to disease than GA. Chuang et al. [CLCY12] also introduced an Odds ratio-based Genetic Algorithm to predict SNP-SNP interactions in breast cancer data. The authors worked on simulated dataset and concluded that this method can apply on other SNP datasets as well; based on statistical and computational analysis, it is also mentioned that the proposed algorithm can efficiently perform on other association studies. An adequate review is done by Upstill-Goddard, Eccles, Fliege, and Collins [UGEFC12]

on machine learning approaches for identifying gene-gene interactions. In this study, popular machine learning methods, which used for reducing irrelevant features, are studied; for instance, Multifactor-Dimensionality Reduction (MDR), Neural Networks (NN), Random Forest, and Support Vector Machines (SVM) are evaluated and their strengths and limitations are presented. Furthermore, in the same study, McKinney, Reif, Ritchie, and Moore [MRRM06] surveyed machine learning methods employed for detecting gene-gene interactions. In this work, not only the popular machine learning methods, which studied in the prior survey, are evaluated, also Cellular Automata (CA) is studied; the authors analyzed and assessed the various machine learning methods in detecting gene-gene interactions and provided proper interpretations about them.

There are a lot of related studies which have considered various feature selection methods on genetic datasets. Some of them concentrate on the classification and diagnosis of a disease, while using feature selection methods. The others focus on studying different feature selection methods directly in this area. Generally, different feature selection approaches addressed in this section show that how important is extracting useful knowledge from genetic data,so clearly excellent feature selection process is needed to this end. However, in all the mentioned studies, the feature selection methods do not choose all the relevant features to phenotype/disease.

In this thesis, we show that under complexity assumptions (and in some models unconditionally) it is not possible to solve feature selection problem in SNP genetic datasets in polynomial time. Thus, any heuristic can be expected to lose accuracy as the number of relevant features grows.

## 2.2 Feature selection heuristics

In this section, we offer a general introduction to the current popular heuristic feature selection methods which are mostly used for genetic data. Besides this, we will analyze these heuristic methods based on their results, mechanisms, popularity, and other exclusive aspects.

Data scientists apply different popular and beneficial feature selection methods on genetic datasets. This variety of methods is based on the type of data and different researchers' viewpoints. Different methods are used in this area, such as Support Vector Machines (SVMs), Evolutionary Algorithms, Neural Networks, logistic regression, odds-ratio, relief algorithms etc. In addition, multi-stage and hybrid methods, which are basically a combination of different methods, are employed to reduce the size of genetic datasets. Although researchers mostly address the common categorizations (filter, wrapper, and embedded) of feature selection methods, certain researchers classify different classes for feature selection methods. For example, exhaustive search, heuristic search, and hybrid methods are used to classify different classes for feature selection methods [WWC16]. Indeed, these heuristic feature selection methods do not solve the problem, they only alternatively give a proper solution, which is the nature of heuristics. This happens because most of the genetic datasets include a small number of samples or individuals with numerous features, which makes the feature selection problem hard to solve. In this study, feature selection methods, which apply to genetic data, are studied. These feature selection methods are categorized to mainly three issues of [LPH+13]:

1. Identification of relevant features for determined diseases.
2. Classification based on samples with recognized disease class labels.
3. Classification of the unknown samples into known disease classes.

This thesis is concentrated on the first category, which can be called case-control

studies or even XOR problems if we specifically work on SNP data [MW07, Hua15].

Finding a proper feature selection with the purpose of increasing accuracy and reducing complexity, is not an easy task [BCSMAB13]. In most cases, the authors of prior work introduce one or more feature selection methods in their works with particular dataset, then demonstrate that their feature selection method is beneficial compared to the other methods which are mentioned in their articles. In [HG15], the authors reviewed different feature selection methods on micro-array data. In this work, the most popular and significant feature selection methods are compared to each other. These methods include t-test feature selection [JA06], Correlation-based feature selection (CFS) [Hal99], Bayesian networks [HHE04, RJFD10],Information gain (IG) [YZZZ10],Genetic algorithms (GA) [JUA05,OT03],Sequential search, SVM method of recursive feature elimination (RFE) [GWBV02], Random forests [DUDA06, JDC+04], and Least absolute shrinkage with selection operator (LASSO) [MSH07]. The feature selection methods are compared based on their types, being supervised or unsupervised, being linear or nonlinear, and their mechanisms [HG15]. Also, the authors [HG15] discussed that using prior knowledge of different biological sources increases the accuracy and decreases the cost of computational complexity of feature selection methods. In comparison with four feature selection methods, which are t-test, significance analysis of micro-arrays (SAM), rank products (RP), and random forest, SAM gained the best performance [LPH+13]. Also, Relief family algorithms are extremely suitable in feature selection problems' particularly for XOR problem types, such as identifying interacting SNPs or features [Hua15]. These algorithms are classified as pair-wise feature ranking methods [Hua15]. The Relief family with their low computational cost are the proper methods for SNP datasets with a huge number of features. Furthermore, Relief algorithms are popular for use in noisy datasets with a huge number of features [MW07]. Due to the power of Relief family methods

(low complexity and robust against noise), these methods are a great choice for hybrid methods [İT13b, SI07, YWCY17, CLCY12, YMLC16, LGR+07]. These methods are used for those hybrid feature selection methods which generally remove irrelevant features in different steps. Suitable algorithms for XOR problems, such as relief algorithms which have low computational complexity, frequently perform solidly [Hua15] as the first step of hybrid methods which reduce most parts of irrelevant features.

These heuristic feature selection methods are employed to work on different real and synthetic datasets. In this study, we study synthetic datasets produced by a software called GAMETES which will be described in detail in the following section.

## 2.3 Synthetic datasets

In the case of synthetic data, the performance of the feature selection methods clearly depends on the learning method operation which is subsequently employed , and it can clearly differ for every method. Additionally, the evaluation of feature selection achievement can be gained by various metrics like computer resources(memory and time), accuracy, ratio of features selected, etc. Also, data scientists may face several challenges such as multiple class output, noisy data, a huge number of irrelevant features, redundant or repeated features, etc. [BCSMAB13].

To obtain synthetic datasets which are very similar to real ones, we employed GAMETES datasets with high dimensional features (low dimensional samples), and noisy data models. These are the problems which data scientists consider when they work on genetic data. In the following, the GAMETES software will be introduced and all necessary information to accurately understand the software is presented.

### 2.3.1 GAMETES

Genetic Architecture Model Emulator for Testing and Evaluating Software (GAMETES) is a software package to generate complex biallelic Single Nucleotide Polymorphism (SNP) disease models for simulation studies [UKSA+12]. This software produces strict and pure n-locus models which are provided under certain significant factors in genetics. In GAMETES, it is possible to create models by defining its Minor Allele Frequency (MAF), heritability of SNPs, and population prevalence [UKSA+12]. In this section, first we describe the information needed to understand the genetic terms and concepts used in GAMETES models, then the mechanism of GAMETES is introduced. Finally the feature selection problem, which we concentrate on, will be discussed at the end of the section.

Single nucleotide polymorphisms (SNPs") are single loci in the DNA sequence and frequently found in the DNA between genes; SNPs can interchange nucleotides (i.e. alleles). Nearly all identified SNPs are biallelic which includes just two alleles of minor(a) or major(A) in a population. Also SNPs can assist scientists in locating genes that are associated with a specific disease because they play a part as biological markers. Three genotypes, $AA$, $Aa$, $aa$, are the possible possessions that a biallelic SNP can take one of them [UKSA+12]. One more concept which exists in this area is *epistasis*. In general, epistasis, which can influence phenotype, is the interaction between different genes; in the case that we studied, it is quite important for studying complex traits such as diabetes, asthma, and hypertension. The existence of epistasis is a special cause for attention, considering that if the effect of one locus is changed or masked by effects at another locus, then detecting the first locus becomes more complicated and explanation of the combined effects at two loci will be more problematic by their interaction. Subsequently, interactions of more than two loci make the situation more difficult [Cor02]. In the GAMETES study, the authors consider

statistical epistasis, which is the basic description of being a deviation from additivity in a mathematical model, to explain the relationship between multi-locus genotypes and phenotypic variation in a population [Fis19, UKSA$^+$12]. Furthermore, the main focus in GAMETES study is on statistical epistasis which is both strict and pure. If all $n$ loci, but not less than $n$ loci, are predictive of disease status in an n-locus model, it is a pure and strict epistatic model. It should be mentioned that the loci in these models are "fully masked", which means that no predictive information is obtained up to all n loci bring together [UKSA$^+$12]. the GAMETES generates deterministic n-locus models based on a set of random parameters and indicated values of heritability, and minor allele frequencies (MAFs); the GAMETES tries to produce a population based on the restrictions of models. The GAMETES' authors showed that the method they presented, which generates complex genetic models, is fast, reliable, and flexible. [UKSA$^+$12]. Also the method that is employed by the GAMETES benefits the Hardy-Weinberg Law [UKSA$^+$12, HC97] in the role of simulation strategy. Hardy-Weinberg Law states that, if the allele frequencies in a population with two alleles at a locus are $p = f(A)$ (major allele frequency) and $q = f(a)$ (minor allele frequency), then the expected genotype frequencies for $AA$, $Aa$, and $aa$ are $p^2$, $2pq$, and $q^2$, respectively, where $p+q = 1$ . This distribution does not change during different generations until population is in Hardy-Weinberg equilibrium [GT92, UKSA$^+$12]. Another important factor which is used in GAMETES, is *penetrance*. Penetrance is probability of disease based on special genotype or multi-locus genotype (MLG). Penetrance functions or penetrance tables are used in showing the relationship of genetic variation and risk of disease. A simple example of penetrance function of genotypes of one SNP is provided in table 2.1 as follows:

Table 2.1 shows the fully penetrance function that means the phenotype status fully depends on genotypes. In this example, genotype $aa$ with probability 1 shows

| SNP1 | | |
|---|---|---|
| AA(0.25) | Aa (0.5) | aa(0.25) |
| 0 | 0 | 1 |

Table 2.1: Penetrance table(function), showing genotypes of one SNP while $p = q = 0.5$

its main effect on disease (phenotype). In the situation of coping $n$-locus interactions between $n$ loci, penetrance function provides $3^n$ penetrance values based on every $3^n$ multi-locus genotypes such as table 2.2. This table shows the penetrance function of two SNPs which are related to risk of disease with all nine of their possible genotypes. For instance, in Table 2.2, the multi-locus genotype of $(AA - Bb)$ with probability of 100% is predictive to disease, while (aa-bb) has 0% chance to have a disease.

| | SNP2 | | | |
|---|---|---|---|---|
| SNP1 | Genotype | BB(0.25) | Bb (0.5) | bb(0.25) |
| | AA(0.25) | 0 | 1 | 0 |
| | Aa(0.5 ) | 1 | 0 | 1 |
| | aa(0.25) | 0 | 1 | 0 |

Table 2.2: Fully penetrance table(function) for interactions between two SNPs with the genotype frequencies $p1 = p2 = q1 = q2 = 0.5$

#### 2.3.1.1 Statistical epistasis viewpoint in Pure-Strict models

GAMETES produces pure and strict genetic models for showing the interactions between SNPs. These classification concepts are defined based on statistical viewpoint. Pure epistasis means that no one of the interacting loci (individually) have the main effect on disease status; however, there exist one or more multi-locus subsets of them which have main effect on disease. Strict epistasis happens when $n$ loci have an effect on disease (phenotype); however, there is no "$multi - locus$" subset of them which is predictive of disease [UKSA+12]. Thus, a strict-pure model is a type of epistasis

model in which all $n$ locus together have the main effect on phenotype; in contrast, no subset of them or none of them individually is predictive of phenotype. This is the worst case model to distinguish, and is a reasonably realistic model which is generated by the GAMETES. If marginal penetrances are added to table 2.2, then it is possible to find a simple statistical strict-pure model. Table 2.3 shows a simple example of fully penetrant function of a pure and strict model. The fully penetrant function ex-

| | SNP2 | | | | |
|---|---|---|---|---|---|
| SNP1 | Genotype | BB(0.25) | Bb (0.5) | bb(0.25) | Marginal penetrance |
| | AA(0.25) | 0 | 1 | 0 | 0.5 |
| | Aa(0.5 ) | 1 | 0 | 1 | 0.5 |
| | aa(0.25) | 0 | 1 | 0 | 0.5 |
| | Marginal penetrance | 0.5 | 0.5 | 0.5 | $K$ =0.5(prevalence) |

Table 2.3: Pure-Strict fully penetrant table (function) for interactions between two SNPs with their marginal penetrances

amples such as tables 2.1, 2.2, and 2.3 do not illustrate the realistic epistasis as they are quite easy to detect and can show the genotypes which are completely predictive or non-predictive (0 and 1) to disease; however, they are only proper instances to understand the strict-pure models employed in GAMETES. Regarding marginal penetrance, which are significant factors in the process of generating pure-strict models, they are the reasons for creating strict-pure models. For example, in Table 2.3, if a person gains genotype $Aa$ and the genotype of SNP2 is disregarded, the probability of having a disease for this person, under this condition, is calculated below:

$(1 \times 0.25) + (0 \times 0.5) + (1 \times 0.25) = 0.5$

Based on this calculation, the marginal penetrance associated with genotype $Aa$ is 0.5. According to table 2.3, when SNP2 is ignored, the probability of having disease for both genotypes $AA$ and $aa$ are 0.5. Therefore, SNP1's genotype separately is not predictive of disease. The same situation happens for SNP2; all the marginals are the same amounts (0.5), and there are different probabilities for its genotypes.

In this way, SNP2' genotype is not solely predictive of disease either. This is the mathematical definition of strict-pure model, in which all the marginal penetrances are the same while there exist different probabilities for genotypes alone. This similar typical value for all the marginals causes the population prevalence of disease($K$) [UKSA+12]. Realistic models, such as the models that are generated by GAMETES, obtain continuous probabilities between 0 and 1; although table 2.3 presents a pure and strict model, it is hardly a realistic model. In this way, table 2.4 shows an example of pure-strict model which is generally produced by GAMETES.

| | SNP2 | | | | |
|---|---|---|---|---|---|
| SNP1 | Genotype | BB(0.5) | Bb (0.41) | bb(0.085) | Marginal penetrance |
| | AA(0.5) | 0.6417 | 0.2014 | 0.5585 | 0.45 |
| | Aa(0.41) | 0.2269 | 0.7571 | 0.3236 | 0.45 |
| | aa(0.09) | 0.4851 | 0.4078 | 0.4866 | 0.45 |
| | Marginal penetrance | 0.45 | 0.45 | 0.45 | $K = 0.45$(prevalence) |

Table 2.4: Pure-Strict penetrance function for interactions between 2 SNPs ($q1 = 0.3, q2 = 0.29$)

Table 2.4 is an excellent instance of GAMETES model because the entire probabilities of genotypes are not certain (0 or 1), thus it is not easy to detect; moreover, all the marginal probabilities (penetrances) are equal based on the pure-strict definition. The following calculations show the process of calculating the marginal penetrances for table 2.4:

$(0.6417 * \mathbf{0.5})+(0.2014 * \mathbf{0.41})+(0.5585 * \mathbf{0.085})=0.45$

$(0.2269 * \mathbf{0.5})+(0.7571 * \mathbf{0.41})+(0.3236 * \mathbf{0.085})=0.45$

$(0.4851 * \mathbf{0.5})+(0.4078 * \mathbf{0.41})+(0.4866 * \mathbf{0.085})=0.45$

$(0.6417 * \mathbf{0.5})+(0.2269 * \mathbf{0.41})+(0.4851 * \mathbf{0.09})=0.45$

$(0.2014 * \mathbf{0.5})+(0.7571 * \mathbf{0.41})+(0.4078 * \mathbf{0.09})=0.45$

$(0.5585 * \mathbf{0.5}) + (0.3236 * \mathbf{0.41}) + (0.4866 * \mathbf{0.09}) = 0.45$

At this point, the general idea of creating pure-strict models, which are generated by the GAMETES software, is introduced. In addition, the detail of methods which GAMETES employs to produce random pure-strict models, such as filling parameters of models, solving penetrance functions based on random initialized parameters, etc. exist in the official GAMETES paper [UKSA+12]. In our study, the most important point of the GAMETES is understanding the main idea of strict-pure models, which leads to a hard problem to solve in the feature selection of the GAMETES datasets. In the following chapters, the relation of pure-strict models and feature selection in datasets which are produced by the GAMETES software is described.

# Chapter 3

# Learning Theory and Complexity Background

Learning theory is one of the theoretical areas of computer science that studies the methods of designing programs which can learn and identify the computational limits of learning by machines. In this area, researchers attempt to evaluate the learning algorithms based on their performance on different problems; however, providing a significant comparison between learning algorithms is not straightforward. In this way, learning theory presents a formal structure of particular defining and focusing on challenges which are related to the performance of various learning algorithms. By employing learning theory, it is also possible to examine the predictive capability and computational performance of learning algorithms. Moreover, learning theory models are the reflection of real problems in life. Therefore, this close relation can be supportive in explaining practical performance which is noticeably a close link to the machine learning research area.

In regard to the cryptography's role in learning theory, it could be mentioned that machine learning and cryptanalysis can be considered as *deeply related* study

areas because they share a great portion of identical approaches and subjects [Riv91]. During a regular process of cryptanalysis, a cryptanalyst attempts to identify the secret key used by the users of a cryptosystem, where the general system is already known. In other words, the cryptanalyst tries to break the cryptosystem. Known functions, those are indexed by the key, generate the decryption function, and the cryptanalyst's job is to find the accurate function which is being used. This is the problem of "learning an unknown function" (the decryption function) from instances of its input behavior [Riv91].

Prior to this, in the Boolean domain there are important problems in learning theory such as learning disjunctions of terms over $\{0, 1\}$-valued variables. [Kat07, FGKP06, O'D14]

In the following sections, we will define some of the core concepts of computational learning theory that we use in this thesis such as PAC learning model, learning parity with and without noise, $k$-juntas, and statistical queries.

## 3.1 PAC learning model

The fountainhead of computational learning theory was Leslie Valiant's 1984 paper [Val84] in which he defined a learning model which became known as PAC learning. In the PAC (probably approximately correct) model, the learner is presented with labeled examples of the output of an unknown function $f$ from some concept class $C$, where the examples are generated according to some distribution $D$. In this model, we have two other inputs: $\epsilon$ and $\delta$; $\epsilon$ is the error and $\delta$ is a probability value in achieving an accuracy. The goal of the learner is producing a hypothesis ($h$) for that function ($f$) which, with probability $\delta$, will be correct with probability at least $1 - \epsilon$ on samples drawn from $D$. A concept class $C$ is $(\epsilon, \delta)$-PAC-learnable if there is a

learner which can learn every function from $C$ in this sense, for any distribution $D$. In this model, the hypothesis $h$ is a function which is close to an unknown function $f$. More formally, the following is the definition of PAC learning:

**Definition 3.1.1.** *Let $C$ be a concept class of Boolean functions, and $\epsilon, \delta$ constants, $0 < \epsilon, \delta \leq 1$. Then $C$ is $(\epsilon, \delta)$-PAC learnable if there is an algorithm $A$ which for every $f \in C$, given as inputs $\epsilon$, $\delta$, and a set of random examples selected from any probability distribution $D$, outputs $h$ (hypothesis) such that with probability at least $\delta$, $Pr_{x \sim D}(h(x) \neq f(x)) \leq \epsilon$.*

For example, in [Val84] Valiant presented a polynomial-time algorithm that learns $k$-CNF formulas in this sense; there, he also pointed out that coping with irrelevant attributes is important. In later work, algorithms are provided for monotone DNF formulas. Moreover, Valiant presented cryptographic evidence that Boolean circuits are not learnable [Ang92].

### 3.1.1 Noise models

In the real world of machine learning, data scientists regularly cope with noisy data. It is not surprising, then, that there has been a lot of work on learning in the presence of noise, and on developing noise-tolerant algorithms [BKW03, O'D14, Riv91, KMV08, SB13]. In the basic PAC learning, it is assumed that the examples from $D$ are correctly labeled by a specific function from the concept class that we are trying to learn; however, in practical work, the training data is noisy, and the labels may not always be correct. To address that, several generalizations of the PAC learning model to the noisy scenarios have been developed, with different noise distributions including Random Classification Noise, Malicious Classification Noise, Uniform Random Attribute Noise, Product Random Attribute Noise, and Malicious Noise [AL88, Slo88, GS95, Val85].

## 3.2 Statistical Query Model

In the PAC learning model, the learner has access to random labeled examples. There has been a flurry of papers after PAC learning was defined that presented variants of PAC learning with different types of access to the data. For example, rather than accessing a random example, a learner could ask for a label on a specific string: this gave the stronger membership query model. Alternatively, instead of being able to access examples with their labels, the learner might be able to gain just some statistics about the data. Moreover, this statistics may be imprecise, and modeling even more closely to the noisy real life applications.

To formalize learning in this latter setting, Kearns [Kea98] introduced the statistical query model (SQ). In this case, the learner asks queries of the form "what is the probability (with respect to $D$) of examples having the property $\phi(x, l)$, where $l = f(x)$ is the label. Moreover, the answer to the query is imprecise, with an allowed additive error $\alpha$. In other words, SQ is a restricted version of PAC learning model, in which a learning algorithm can acquire estimates of statistical traits of the examples, however, there is no access to the examples themselves [Fel12]. More formally, the definition of SQ is as follows:

**Definition 3.2.1.** *Fix an input space $X$, a function $f$ and a probability distribution $D$. A* statistical query *with tolerance $\tau$ for given $(f, D)$ takes as an input $(\phi, \tau)$, where $\phi\colon X \times \{0,1\} \to \{0,1\}$, and returns a value $v$ such that*

$$Pr_{x\sim D}(\phi(x, f(x)) = 1) - \tau \leq v \leq Pr_{x\sim D}(\phi(x, f(x)) = 1) + \tau.$$

*Let $C$ be a concept class of Boolean functions. Then an algorithm $A$ learns $C$ in the statistical query model if, for every $f \in C$, distribution $D$, $A$ outputs a hypothesis $h$ which, with probability $\delta$ agrees with $f$ with probability at least $1 - \epsilon$, where $A$*

*learns about $f$ by asking a sequence of statistical queries, with no access to individual samples.*

## 3.3 Learning Parity with Noise (LPN)

The core of our hardness result is a reduction from the Learning Parity with Noise (LPN) problem. In this section we define LPN and survey relevant results about it.

Learning Parity is a well-studied problem in computational learning theory, complexity theory and machine learning. There, the input to the algorithm is a list of $m$ labeled samples $\{(x_1, y_1), \ldots, (x_m, y_m)$ from some distribution $D$, where for all $i$ $x_i \in \{0,1\}^n$ and $y_i \in \{0,1\}$. The learning algorithm has to determine the labeling function $f\colon \{0,1\}^n \to \{0,1\}$ under the promise that $\forall i, y_i = f(x_i) = \langle x_i, s\rangle$, where $s \in \{0,1\}^n$ is a fixed "secret key" (that is, $f(x_i)$ is the parity of bits of $x_i$ in locations where $s$ has a 1; for each instance of the learning parity problem, the answer is fully determined by $s$). A parity is $k$-sparse when $s$ contains at most $k$ 1s.

The Learning Parity problem does have a polynomial-time algorithm: it can be solved by Gaussian elimination. However, it is a classic example of a function that is very far from being linearly separable and thus is not learnable by a perceptron even for $k = 2$. [MP69]

A harder version of this problem is *Learning Parity with Noise (LPN)*, in which samples may contain some noise [BKW03, KMV08, Reg09, Pie12, KMV08]. Here, we follow the definition from [Reg09, LF06, GJL14, FMI+06]:

**Definition 3.3.1** (Learning Parity with Noise (LPN))**.** *An instance of Learning Parity with Noise (LPN) is a list of $m$ samples $x \in \{0,1\}^n$, chosen uniformly at random, and each sample is labeled with $y \in \{0,1\}$, where $y$ is the inner product of $x$ and secret key ($s$) which is a fixed vector $s \in \{0,1\}^n$; now, $y = \langle x, s\rangle$ with probability*

*of* $1-\epsilon$ *(noise), otherwise* $y = 1 - \langle x, s\rangle$*, for the secret key* $s$*. The instance is* $k$*-sparse if* $s$ *has at most* $k$ *1s. An algorithm solves this problem if given as an input* $(x_1, y_1)\ldots(x_m, y_m)$ *produced as described it returns* $s$ *with probability at least* $\theta$ *for some constant* $\theta$ *[GJL14, Pie12].*

It has been conjectured that LPN is hard on average. Based on this assumption, Pietrzak [Pie12] introduced a number of cryptographic primitives using simple LPN based schemes, in particular pseudorandom generators and symmetric key encryption over secret-key authentication protocols, as well as public key identification, commitments, and zero-knowledge proofs. A more general version of the problem where the domain is not Boolean, called learning with errors [Reg10], has been used as a basis for even more powerful cryptography. The best algorithm for LPN to date based on Gregory Valiant's [Val12], is achieving complexity of $O(n^{\frac{\omega+\epsilon}{3}}k)$, where $\omega$ is the complexity of matrix multiplication.

While cryptographic results are based on the assumption of average-case hardness of LPN, Kearns [Kea98] has shown unconditionally that noisy parity cannot be learned with polynomially many queries in the Statistical Query model. In that model, an algorithm can only ask queries of the form "what fraction of strings satisfying such-and-such condition have a label y?" and, moreover, the answer it receives can have an additive error up to a tolerance bound. As the Statistical Query model is equivalent to some differential privacy frameworks, this makes LPN not learnable in the corresponding differentially private frameworks.

In the PAC model, Blum, Kalai, and Wasserman [BKW03] presented a slightly sub-exponential time algorithm for learning parity functions in the presence of random noise. This article [BKW03], is started with a key question in machine learning: "What kinds of functions can be learned efficiently from noisy, imperfect data?" Feldman, Gopalan, Khot, and Ponnuswami [FGKP06] presented well-studied problems

concerning the learnability of parities and half-spaces in the presence of classification noise. This study illustrated that, under the uniform distribution, learning parities with adversarial classification noise reduces to learning parities with random classification noise. The authors presented two different reductions to LPN under uniform distribution; this article supports the belief that learning noisy parity under the uniform distribution, is a hard problem. Moreover, the authors, based on [BB02], proved an essential optimal hardness factor for half-space of which majorities are hard to PAC-learn (Probably Approximately Correct) using any representation, based on the cryptographic assumption underlying the Ajtai-Dwork cryptosystem [FGKP06]. In [KMV08], with a suitable definition of the agnostic weak learner, it is demonstrated that the boosting by branching programs algorithm can be analyzed in the agnostic setting; also, the authors showed the utility of this fact in the first nontrivial algorithm. Lyubashevsky [Lyu05] proved that there is an algorithm which solves the parity problem in the presence of noise in time $2^{O(n/loglogn)}$. This work is a response to a key question of Blum's, Kalai's, and Wasserman's [BKW03] key question which was "is there any $2^{o(n)}$ algorithm for the length-$n$ parity problem that uses only $poly(n)$ labeled examples?" This paper [Lyu05] also presented a sub-exponential algorithm for decoding random linear codes, and extended the same techniques to support sub-exponential algorithm for dense instances of the random subset sum problem.

Katz [Kat07] discussed re-casting LPN as the problem of decoding a random linear code, as a possible tool for developing highly-efficient cryptographic primitives. This study reviewed recent work with the goal of creating efficient authentication protocols based on the conjectured hardness of LPN. There, Katz also considered an efficient protocol based on the LPN problem that is provably resistant to man-in-the-middle attacks with an open question that whether the LPN problem can be employed to construct efficient cryptographic protocols for other tasks. The author cited Regev's

result [Reg09] to show that public-key encryption can be based on a generalization of the LPN problem.

Learning parity is a special case of learning $k$-juntas, where the labeling function, while still dependent on $k$ bits of its input, is not necessarily a parity function. There, the learning algorithm has to determine both the relevant $k$ bits and which function $f$ of those $k$ bits is being computed [DMN18, ABR16].

Bhattacharyya, Gadekar, Ghoshal, and Saket [BGGS15] proved that the learning sparse parities is hard as the W[1]-hardness holds for learning a k-parity using a k-junta. The following is their theorem [BGGS15]:

**Theorem 3.3.1.** *[BGGS15] The following is W[1]-hard: for any constant $\delta > 0$, given $m = O(k \cdot 2^3 k \cdot (\log n)/\delta^3)$ point-value pairs $\{(z_i, b_i)\}_{i=1}^m \subseteq \mathbb{F}_2^n \times \mathbb{F}_2$, decide whether:*

*YES Case: There exists a k-parity which satisfies all the point-value pairs.*

*NO Case. Any function $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2$ depending on at most k variables satisfies at most $\frac{1}{2} + \delta$ fraction of the point value pairs.*

That is, unless $W$-hierarchy collapses, learning $k$-juntas even approximately requires time $O(n^{\Omega(f(k)}$ for some $f(k)$. Note that learning $k$-juntas exactly is $W[2]$-hard [AKL09].

If membership queries are available, Aliakbarpour, Blais, and Rubinfeld [ABR16] illustrated that it is possible to learn k-juntas with respect to the uniform distribution over the Boolean hypercube. The authors obtained the results by a new Fourier-based learning algorithm inspired by the Low-Degree Algorithm of Linial, Mansour, and Nisan [LMN93]. This study also provided a nearly-optimal algorithm for verifying that an unknown distribution is a k-junta distribution with respect to the uniform distribution. Subsequently, they determined the connections of k-junta distributions and testing uniformity of weighted collections of distributions. In another study,

Blais [Bla10] has shown that it is possible to test whether a given distribution comes from a $k$-junta when membership queries are available.

# Chapter 4

# Our results

In this chapter, we present the results of our research based on studies we have completed on the feature selection of pure and strict datasets, which are produced by GAMETES software [UKSA+12]. We will show that solving the GAMETES problem is computationally hard under the variant of a cryptographic assumption. Namely, GAMETES can produce models in such a way that solving feature selection in datasets generated from these models is at least as hard as solving the $k$-sparse LPN problem defined in the previous chapter.

## 4.1 The GAMETES problem

Let us first formally define the problem of feature selection in GAMETES datasets. We will define the notion of a "GAMETES model" used to generate a dataset. The GAMETES problem itself will then be solving feature selection on a dataset produced by a GAMETES model.

Let $X = \{x_1, \ldots, x_m\}$ be a set of strings of length $n$ (here, the strings are over the alphabet $\{0, 1, 2\}$). Let $J = \{j_1, \ldots, j_k\}$ be a set of indices in the range of $[1, \ldots, n]$. Now, for a given string $x$, denote by $x[J]$ its restriction to the indices in $J$, that is,

the $k$-array string $x[j_1] \dots x_i[j_k]$. Finally, let $X \downarrow_{x[J]=s}$ denote the set of all strings in the database such that $x[J] = s$.

A *GAMETES model* is defined by the following list of parameters. The descriptions of the parameters below are mostly from the supplementary materials to [UKSA+12], with some simplifications.

- $k$: The number of relevant SNPs (that is, SNPs which together are correlated with the outcome).

- $q$: Minor allele frequency. Each SNP can have three values: 0 (both alleles are dominant) which occurs with probability $(1-q)^2$; 1 (one dominant, one recessive) which occurs with probability $2q(1-q)$; and 2 (both are recessive) which occurs with probability $q^2$. Given $q$, let $D_q^\ell$ be the distribution of ternary strings $w$ of length $\ell$ where each index in $w$ is sampled independently according to probabilities above.

  In general, GAMETES can produce models with separate minor allele frequencies $q_1, \dots, q_k$ for every relevant SNP, plus $q_r$ for every non-relevant SNP. As it is easy to determine (by counting) which features are irrelevant if $q_r$ is different from $q_1, \dots, q_k$, we will focus on the case when all $q_i$ are the same.

- $K$: prevalence, or population (marginal) penetrance, defined as the expectation of a positive outcome over all possible gene sequences (weighted by their frequencies according to $q$). That is, if $\mathcal{G}$ is a set of all possible $3^k$ strings in $\{0,1,2\}^k$, $p(x)$ is a probability of a string $x$, and $f(x)$ is the probability of a positive outcome given string $x$, then $K = E_{x \in \mathcal{G}} p(x) f(x)$.

- $h$: heritability. This is essentially a normalized standard deviation of the out-

come. More specifically, [UKSA$^+$12] defines heritability by the following

$$h^2 = \frac{1}{K(1-K)} \Sigma_{x \in \mathcal{G}} p(x)(f(x) - K)^2 \tag{4.1}$$

Note that if the outcome is chosen uniformly at random with probability $K$, then $h = 0$, because in this case $f(x) = K$ for all $x$. If the outcome is fully determined (that is, $f(x) = 1$ or $f(x) = 0$ for all $x$), $h = 1$. Therefore, for feature selection to make sense, $h$ should be significantly greater than 0, while still at most 1.

**Definition 4.1.1** (A GAMETES model)**.** *Let $k$ be the number of relevant features, $q$ the minor allele frequency, $K$ the prevalence, and $h$ heritability. A* GAMETES model *with parameters $k, q, K, h$, also called a penetrance function, is a $k$-dimensional array $M(w[1], \ldots, w[k])$, where each dimension has 3 possible values ($3^k$ values total, indexed by strings over $\{0,1,2\}^k$), with 0, 1, 2 encoding the number of recessive alleles in a SNP). For all $w \in \{0,1,2\}^k$, $0 \leq M(w) \leq 1$ denotes the probability of the positive outcome (phenotype) given a genotype with relevant SNPs denoted by $w$. Note that in [UKSA$^+$12], $M(w)$ is denoted by $f_{i_1 \ldots i_k}$, where $i_1 \ldots i_k = w$.*

*A model is pure and strict if for every $j$, $1 \leq j \leq k$, and every combination of values of SNPs other than the $j$'th SNP, the following equation is satisfied*

$$(1-q)^2 \cdot M(w_{w[j] \leftarrow 0}) + 2q(1-q) \cdot M(w_{w[j] \leftarrow 1}) + q^2 \cdot M(w_{w[j] \leftarrow 2}) = K.$$

*Here, the notation $w_{w[j] \leftarrow b}$ denotes $w$ with $j^{th}$ bit set to $b$. That is, unless the values of all $k$-relevant SNPs are known, the probability of any outcome is its marginal probability, $K$ (penetrance). Finally, heritability of the model is $h$ as computed from equation 4.1, with probabilities of strings according to $D_q^k$.*

There is a long discussion in [UKSA+12] about different ways of generating such penetrance functions. What is important to us is that they seem to be able to produce any model which satisfies the conditions of being pure and strict, with penetrance and heritability within their respective bounds as a function of $q$. In particular, for $q = 1 - 1/\sqrt{2}$ they can generate models with any heritability up to 1. Therefore, for the rest of the discussion, we will equate the set of penetrance functions (models) $M$ as described above with the set of models that GAMETES software can produce.

Given GAMETES model, the GAMETES problem can be defined as identifying $k$ features generated by a GAMETES model in a given dataset of $m$ labeled samples on $n$ features, where the rest of the features are generated randomly with frequencies of $0, 1$, or 2 according to $q$.

**Definition 4.1.2** (The GAMETES Problem)**.** *Let $n$ be the total number of features, and $m$ the number of samples. Denote a positive outcome (i.e. a case) by $+$, and a negative outcome (control) by $-$. The* GAMETES problem*, with parameters $k, q, K, h, n$, denoted by $GAMETES(k, q, K, h, n, m)$ is defined as follows:*

**Input:** *A pair $(X, Y)$ where $X$ consists of $m$ strings $x_1, \ldots, x_m$, where for every $i$, $x_i \in \{0, 1, 2\}^n$, and $Y$ consists of $y_1, \ldots, y_m$ corresponding labels, $y_i \in \{-, +\}$.*

**Promise:** *Strings in $X$ are generated uniformly at random with frequencies of $0, 1, 2$ according to $(1-q)^2, 2q(1-q)$, and $q^2$ respectively (that is, sampled from $D_q^n$). There is a set $J$ containing $k$ indices and a GAMETES model $M$ with parameters $k, q, K, h$ generating $Y$ from $X$ so that for each $i$, $Pr(y_i = +) = M(x_i[J])$.*

**Output:** *$J = \{j_1, \ldots, j_k\}$ such that $X[J]$ is strongly correlated with the outcome. In particular, when $m >> k$, then based on heritability 4.1, the following relationship*

*occurs:*

$$\sum_{w \in \{0,1,2\}^k} \frac{|X \downarrow_{x[J]=w}|}{m} (Pr(y_i = + \mid x[J] = w) - K)^2 \approx h^2 \cdot K \cdot (1 - K)$$

## 4.2 Reducing $k$-sparse LPN to the GAMETES problem.

The central technical result of this section is the reduction from $k$-sparse LPN to the problem of feature selection in a dataset generated from a pure strict model with $k$ relevant features out of $n$ total features (the GAMETES problem).

First, let us consider a toy example, to illustrate the connection between LPN and the GAMETES problem. Imagine a binary version of the GAMETES problem: that is, a version generating datasets with "binary genotypes", which are strings over $\{0, 1\}$ as opposed to $\{0, 1, 2\}$. In this case, the parity function can be encoded by the model $M(b_1, \ldots, b_k) = b_1 \oplus \cdots \oplus b_k$. Here, there is no analogue to $q$, and probabilities of all entries of $M$ are the same. Now, finding the $k$ relevant features on a dataset produced by this model $M$ is exactly the same as determining the parity of which $k$ bits of input strings gives the output.

### 4.2.1 The reduction.

Let $\{(\overline{x_1}, y_1), (\overline{x_2}, y_2), \ldots, (\overline{x_m}, y_m)\}$ be an instance of learning k-sparse parity function (with or without noise). The $x_i$ are binary strings of length $n$ chosen uniformly at random, and $y_i$ are bits. If this is a noiseless scenario, then there exists a string $s$ of

length $n$ with exactly $k$ 1s such that for every $i$,

$$y_i = \langle s, \overline{x_i} \rangle = (\sum_{j=1}^{n} s[j] \cdot x_i[j]) \mod 2.$$

Otherwise, if there is $\epsilon$-noise, $Pr[y_i = 1 - \langle s, \overline{x_i} \rangle] = \epsilon$, and $Pr[y_i = \langle s, \overline{x_i} \rangle] = 1 - \epsilon$.

We will construct the corresponding dataset $(Z, Y) = \{(\overline{z_1}, y_1) \ldots, (\overline{z_m}, y_m)\}$ with $\overline{z_i} \in \{0, 1, 2\}^n$ for all $i$ as follows. Let $q = 1 - 1/\sqrt{2}$. For every bit $x_i[j]$, set $z_i[j] = x_i[j]$, except if $x_i[j] = 1$, then set $z_i[j] = 2$ with a probability of $q^2 = (1.5 - \sqrt{2}) \approx 0.085786$. Note that with this choice of $q$, the probability of $z_i[j]$ being 0 is 1/2, being 1 is $2(1/\sqrt{2})(1 - 1/\sqrt{2}))$, and being 2 is $(1 - 1/\sqrt{2})^2$. This completes the reduction, with $(Z, Y) = \{(\overline{z_1}, y_1) \ldots, (\overline{z_m}, y_m)\}$ being the resulting instance of the GAMETES problem.

For the case of learning parity without noise, $(Z, Y)$ can be produced by GAMETES with the minor allele frequency $q$ and a model $M$ with $M(w) = b_1 \oplus \cdots \oplus b_k$, where $b_j = 0$ if $w[j] = 0$, otherwise (when $w[j] = 1$ or $w[j] = 2$) $b_j = 1$. For example, for $k = 2$ the model will look as follows:

| | SNP2 | | | | |
|---|---|---|---|---|---|
| SNP1 | Genotype | BB(1/2) | Bb $(2(1/\sqrt{2})(1 - 1/\sqrt{2}))$ | bb$((1 - 1/\sqrt{2})^2)$ | Marginal penetrance |
| | AA(1/2 ) | 0 | 1 | 1 | 1/2 |
| | Aa$(2(1/\sqrt{2})(1 - 1/\sqrt{2}))$ | 1 | 0 | 0 | 1/2 |
| | aa$((1 - 1/\sqrt{2})^2)$ | 1 | 0 | 0 | 1/2 |
| | Marginal penetrance | 1/2 | 1/2 | 1/2 | $K$ =1/2(prevalence) |

Table 4.1: The model $M$ for $k$=2 without noise generating the GAMETES dataset produced as the result of the reduction

In the presence of $\epsilon$-noise, that is, when $y_i = b_1 \oplus \cdots \oplus b_k$ with probability $1 - \epsilon$ for $b_j$ as above, and $y_i = 1 - b_1 \oplus \cdots \oplus b_k$ with probability $\epsilon$, let us define $M(w) = Pr[y_i = 1]$, so $M(w) = |b_1 \oplus \cdots \oplus b_k - \epsilon|$. For example, for $k = 2$, the model $M$ is a 2-locus strict-pure penetrance function, as shown below.

| | SNP2 | | | | |
|---|---|---|---|---|---|
| SNP1 | Genotype | BB(1/2) | Bb $(2(1/\sqrt{2})(1-1/\sqrt{2}))$ | bb$((1-1/\sqrt{2})^2)$ | Marginal penetrance |
| | AA(1/2) | $\epsilon$ | $1-\epsilon$ | $1-\epsilon$ | 1/2 |
| | Aa$(2(1/\sqrt{2})(1-1/\sqrt{2}))$ | $1-\epsilon$ | $\epsilon$ | $\epsilon$ | 1/2 |
| | aa$((1-1/\sqrt{2})^2)$ | $1-\epsilon$ | $\epsilon$ | $\epsilon$ | 1/2 |
| | Marginal penetrance | 1/2 | 1/2 | 1/2 | $K = 1/2$(prevalence) |

Table 4.2: The model $M$ for $k$=2 generating the GAMETES dataset produced as the result of the reduction

### 4.2.2 Correctness of the reduction

The following lemma states the reduction more formally, spelling out the parameters.

**Lemma 4.2.1.** *For every instance of k-LPN there exists a GAMETES model which can generate an instance of the GAMETES problem produced by the reduction above. The parameters of this models are k, minor allele frequency* $q = 1-1/\sqrt{2}$*, penetrance* $K = 0.5$ *and heritability* $h = 1 - 2\epsilon$*.*

*Proof.* Let $I = \{(\overline{x_1}, y_1), (\overline{x_2}, y_2), \ldots, (\overline{x_m}, y_m)\}$ be an instance of k-sparse LPN with noise parameter $\epsilon$. That is, for all $i$, $x_i \in \{0,1\}^n, y_i \in \{0,1\}$, and there exists a secret string $s \in \{0,1\}^n$ with exactly $k$ 1s, and for every $i$, $y_i = \bar{x}_i \oplus s \oplus e_i$ where $e_i = 1$ with probability= $\epsilon$. Solving this instance of k-sparse LPN is finding $s$, that is, finding the $k$ positions at which $s$ equals 1. Now let $I' = (Z, Y)$ be constructed as described in the previous subsection; that is, for all $i, j$, if $x_i[j] = 0$ then $z_i[j] = 0$, and if $x_i[j] = 1$, then $z_i[j] = 2$ with probability $q^2 = 1.5 - \sqrt{2}$, and $z_i[j] = 1$ otherwise.

Let $J$ be the set of indices such that $s[j] = 1$. For a string $w \in \{0,1,2\}^k$, let $b_1 \ldots b_k \in \{0,1\}^k$ be the corresponding binary string (that is, $b_j = 0$ when $w[j] = 0$, and $b_j = 1$ otherwise). Now, define a GAMETES model $M$ by $M(w) = 1 - \epsilon$ when $b_1 \oplus \cdots \oplus b_k = 1$, and $M(w) = \epsilon$ otherwise, with $q = 1 - 1/\sqrt{2}$.

The instance $I'$ from the reduction is a possible instance of the GAMETES problem generated by this model with parameters $n$ and $m$. GAMETES first chooses positions of $k$ out of $n$ relevant features randomly; thus, for any secret key $s$ from $k$-LPN

instance, it can choose $k$ positions $J$ to be exactly where $s$ is 1. Then, it generates $(Z, Y)$, where each $z_i[j] = 0$ with probability 1/2 (same as probability of $x_i[j] = 0$. Finally, by construction of $M$, for each $z_i$ $y_i = +$ with probability $1 - \epsilon$ when $b_1 \oplus \cdots \oplus b_k = 1$, and $y_i = +$ with probability $\epsilon$ when $b_1 \oplus \cdots \oplus b_k = 0$.

Since probability of a parity of $k$ bits being 1 over uniformly random strings is 1/2, probability of $y_i = +$ is 1/2, as probability of + changed to − is the same ($\epsilon$) as probability of − changed to +. Therefore, the expected value of $y_i$ is +, thus $K = 0.5$. Respectively, summing over any index $j$ of $M$, if $k$ is even we get

$$(1-q)^2 \cdot \epsilon + 2q(1-q) \cdot (1-\epsilon) + q^2 \cdot (1-\epsilon) = 1/2 \cdot \epsilon + 1/2 \cdot (1-\epsilon) = 1/2$$

If $k$ is odd, we get the same expression with the roles of $\epsilon$ and $1 - \epsilon$ reversed, which also sums to 1/2. Thus, the model is strict and pure: indeed, no subset of parity correlates with the output of a parity, with or without noise.

Now, to obtain heritability $h = 1 - 2\epsilon$, note that when $w$ is sampled from the distribution $D_q^k$, the probability of $M(w) = 1-\epsilon$ is the same as probability of $M(w) = \epsilon$, and equal to 1/2. Thus, with this and $K = 0.5$, the expression for heritability from 4.1 simplifies to

$$h^2 = \frac{1}{K(1-K)} \sum_{x \in \{0,1,2\}^k} p(x)(M(x)-K)^2 = \frac{1}{0.25}(\frac{1}{2}(\epsilon-0.5)^2+\frac{1}{2}(1-\epsilon-0.5)^2) = \frac{(0.5-\epsilon)^2}{0.25}$$

So $h = \sqrt{(0.5-\epsilon)^2/0.25} = 1 - 2\epsilon$. Thus for any value of $\epsilon \in [0, 0.5)$, heritability is within the bounds which GAMETES can handle (see pages 8-9 in [UKSA+12] for details).

□

The lemma above gives the bulk of the correctness of the reduction proof, by

showing how the reduction produces an instance of the GAMETES problem. Now, it remains to argue that solving the resulting GAMETES problem gives an answer to the original instance of $k$-LPN.

As information-theoretic limit on the number of samples necessary to recover the parity is $O(k \log n)$ [BGGS15], we can assume that $m \geq k \log n$ in the following theorem.

**Theorem 4.2.1.** *Suppose that there is a feature selection algorithm, $A$, that correctly determines $k$ relevant features in every instance of GAMETES problem in time $t(n, k, m, h)$. Then $k$-LPN is solvable in time $t(n, k, m, (1 - 2\epsilon)) + O(n, m)$.*

*Proof.* By Lemma 4.2.1, the reduction from section 4.2.1 produces an instance $I'$ of the GAMETES Problem from any instance $I$ of $k$-LPN, with $h = 1 - 2\epsilon$. Note that this reduction makes a single pass over the instance of $k$-LPN, thus producing $I'$ in time $O(n, m)$. Moreover, number of samples and number of bits (features) are preserved by the reduction.

Now, suppose that the algorithm $A$ returns the set $J$ of $k$ relevant features in $I'$. As by construction the relevant $k$ features of the instance of the GAMETES problem are precisely the indices where the secret key $s$ is 1, recovering these features gives us full information about $s$ used to generate $I$, thus solving $k$-LPN.

□

This theorem now allows us to transfer hardness results and assumptions about $k$-LPN to the GAMETES Problem.

## 4.3 Hardness of the GAMETES problem

First, note that as LPN is an NP-hard problem, so is the GAMETES problem, as our reduction is a polynomial-time many-one reduction. In addition to that, we can

get several other hardness results for the GAMETES problem based on hardness of $k$-LPN.

### 4.3.1 Unconditional hardness results from Statistical Query model

Recall that $k$-LPN is unconditionally hard for the Kearn's Statistical Query model [Kea98]. Thus,

**Corollary 4.3.1.** *No heuristic which only relies on approximate statistics about the genetic dataset (as opposed to access to individual samples) can solve the GAMETES problem in polynomial time.*

An important practical application of the statistical query model is in differential privacy: the class of problems learnable with local differentially private algorithms is exactly the class of problems learnable in the statistical query model [KLN$^+$11, BDMN05].

**Corollary 4.3.2.** *The GAMETES problem is not solvable by local differentially private algorithms.*

### 4.3.2 Parameterized complexity

Theorem of Bhattacharyya, Gadekar, Ghoshal, and Saket [BGGS15] stated on page 28 shows that learning a $k$-parities even approximately is W[1]-hard; moreover, learning $k$-juntas exactly is $W[2]$-hard [AKL09]. As the reduction preserves the dependence on $k$, and learning $k$-juntas reduces to learning parities,

**Corollary 4.3.3.** *Unless $W$-hierarchy collapses, GAMETES problem is not in FPT with respect to parameter $k$.*

In [BGGS15], it is shown that *learning a k-sparse solution to a system of linear equations is fixed parameter intractable*. The researchers mentioned that this happened even when three conditions exist:

1- There are only logarithmic number of equations in the number of variables

2- The learning is permitted to be approximate

3- The learning is permitted to produce as hypothesis any function (junta) supported on at most k variables.

## 4.4 Experimental results

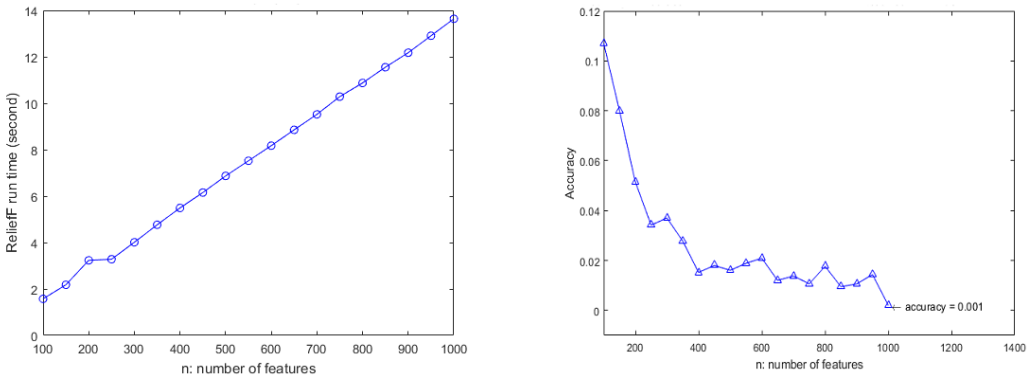
### 4.4.1 Performance of ReliefF

ReliefF is a state-of-the-art heuristic commonly used for feature selection in genetic datasets. Its running time is $O(mna)$ (a is the number of training instances which is the Relief algorithm's configuration input) [UMLC+18], which is polynomial in $n$ and $m$, and does not take $k$ into account. Though for very small $k$ ReliefF performed excellent, its accuracy quickly decreased with increasing $k$. See Figure 4.1 for graphs of the accuracy and performance of ReliefF on GAMETES datasets.

### 4.4.2 An early proposed heuristic

Before we discovered the connection between the GAMETES problem and LPN, we proposed the following heuristic, and evaluated it on the GAMETES datasets. However, as could be predicted from the statistical query lower bounds results, this heuristic does not scale with $k$.

First, the dataset, which is produced by GAMETES, must be split into two categories, case and control, based on the phenotype situation of the dataset. As the




Run-time Accuracy

Figure 4.1: ReliefF run-time and accuracy with fixed k,m and growing n

existing values (genotypes) in SNP datasets are 0s, 1s, and 2s, so in each category, the number of each value for every SNP/feature is counted. Then, the percentage (frequency) of each value for every SNP in both the case and control categories are calculated. In the next step, these percentages are ranked. Now, by defining a function, we measure the significance of each SNP based on the statistics that which is generated. The function is defined below:

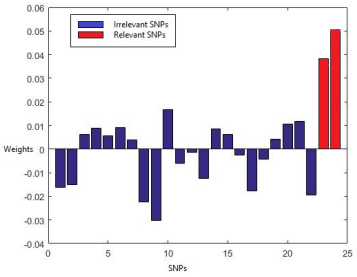
$$f(x) = (\alpha \times V) + (\beta \times P)$$

In this definition, $f$ is the function which calculates the rank of importance of each SNP/feature, $x$ is each SNP/feature, $\alpha$ and $\beta$ are constant numbers which we set them to $\alpha = 50$ and $\beta = 25$; $V$ is a real variable which can be set only to 0 or 1, and $P$ is an important variable in this function which is calculated below:

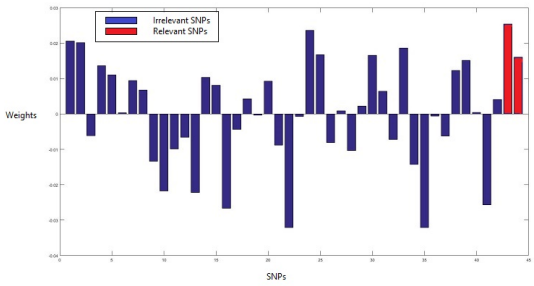
$$if \quad V = 0, P = \frac{-10}{3}((P_{TopRankCase} + P_{TopRankControl}) - 2), \quad otherwise$$
$$P = P_{TopRankCase} + P_{TopRankControl}$$

$P_{TopRankCase}$ and $P_{TopRankCase}$ are the highest frequency or percentage in the case category and control category respectively. If both of the highest frequencies ($P_{TopRankCase}$ and $P_{TopRankControl}$) have the highest frequencies with the same values(genotypes),

then $V$ set to 0, and when they belong to different values(genotypes), then $V$ set to 1. In this procedure, the higher rank obtained for each feature, the more relevant is that feature to the phenotype occurrence. Figure 4.2 is the result of the analysis on one of the pure-strict models of GAMETES which is performed by our statistical method. In this analysis, it is clear that when the total number of features are low, this ranking method returns higher ranks to the relevant features, however by adding just some features, this method is not capable of distinguishing relevant SNPs.




25 SNPs, the last two are correlated




50 SNPs, the last two are correlated

Figure 4.2: Statistical analysis on GAMETES

# Chapter 5

# Conclusions

In this thesis, we worked on the computational hardness of feature selection in strict and pure synthetic genetic datasets. The main contribution of this study is the reduction from the sparse version of Learning Parity with Noise to the GAMETES problem and shows that solving the feature selection in pure-strict datasets, which produced by GAMETES, is computationally hard. In other words, our results proved that solving the feature selection of GAMETES problem is as hard as solving the $k$-sparse LPN. Due to the vital role of feature selection in SNP datasets, and importance of LPN in learning theory and cryptography, finding the hardness relation between these two essential problems is valuable.

Going forward, we are working on extending our results to show that the GAMETES problem, and, moreover, solving feature selection in GAMETES-produced datasets, is hard on average assuming LPN conjecture (that LPN is hard on average for constant noise). It is easy to show that our reduction can work with other values of $K$, by increasing or decreasing probabilities in all cells of $M$ respectively. We would like to show that solving the GAMETES problem for arbitrary feasible GAMETES models is as hard on average as it is to solve it for GAMETES that correspond to

instances from our reduction (that is, models with values other than $\epsilon$, $1-\epsilon$ and with different values of the minor allele frequency $q$). This is a work in progress; we think that we can model general feasible GAMETES models as a "extra-noisy" versions of models that correspond to the reduction from $k$-LPN, and handle different values of $q$ by subsampling from instances produced by the reduction (at some loss of parameters). Ideally, we would like to show that the hardness of the GAMETES problem can be used as a cryptographic assumption on its own. And in addition to that, to make a claim about hardness of feature selection in GAMETES-produced datasets on average we need to verify that the distribution of instances of the GAMETES problem generated by the GAMETES software is close enough to random (ie, that different instances of the GAMETES Problem with the same parameters have a similar chance of being generated).

Another future direction concerns upper bounds. It would be interesting to implement state-of-the-art algorithms for $k$-LPN such as Valiant's algorithm from [Val12], and evaluate its performance on the GAMETES datasets as well as real data.

Finally, this connection between LPN and the GAMETES problem gives an even stronger incentive to study the complexity of $k$-LPN, to give better algorithms for $k$-LPN (and, hopefully, for GAMETES), or prove a $n^{\Omega}(k)$ lower bound.

# Bibliography


[ABR16] Maryam Aliakbarpour, Eric Blais, and Ronitt Rubinfeld. Learning and testing junta distributions. In *COLT*, pages 19–46, 2016.

[AKL09] Vikraman Arvind, Johannes Köbler, and Wolfgang Lindner. Parameterized learnability of juntas. *Theoretical Computer Science*, 410(47-49):4928–4936, 2009.

[AL88] Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988.

[Ang92] Dana Angluin. Computational learning theory: survey and selected bibliography. In *Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*, pages 351–369. ACM, 1992.

[BB02] Nader H Bshouty and Lynn Burroughs. Maximizing agreements and coagnostic learning. In *International Conference on Algorithmic Learning Theory*, pages 83–97. Springer, 2002.

[BBB10] Nahla Barakat, Andrew P Bradley, and Mohamed Nabil H Barakat. Intelligible support vector machines for diagnosis of diabetes mellitus. *IEEE transactions on information technology in biomedicine*, 14(4):1114–1120, 2010.

[BCSMAB13] Verónica Bolón-Canedo, Noelia Sánchez-Maroño, and Amparo Alonso-Betanzos. A review of feature selection methods on synthetic data. *Knowledge and information systems*, 34(3):483–519, 2013.

[BDMN05] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the sulq framework. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 128–138. ACM, 2005.

[BG16] Aicha Boutorh and Ahmed Guessoum. Complex diseases snp selection and classification by hybrid association rule mining and artificial neural network—based evolutionary algorithms. *Engineering Applications of Artificial Intelligence*, 51:58–70, 2016.

[BGGS15] Arnab Bhattacharyya, Ameet Gadekar, Suprovat Ghoshal, and Rishi Saket. On the hardness of learning sparse parities. *arXiv preprint arXiv:1511.08270*, 2015.

[BHOP10] Hyo-Jeong Ban, Jee Yeon Heo, Kyung-Soo Oh, and Keun-Joon Park. Identification of type 2 diabetes-associated combination of snps using support vector machine. *BMC genetics*, 11(1):26, 2010.

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM (JACM)*, 50(4):506–519, 2003.

[Bla10] Eric Blais. Testing juntas: a brief survey. In *Property testing*, pages 32–40. Springer, 2010.

[CEK17] Ramalingaswamy Cheruku, Damodar Reddy Edla, and Venkatanareshbabu Kuppili. Sm-ruleminer: Spider monkey based

rule miner using novel fitness function for diabetes classification. *Computers in biology and medicine*, 81:79–92, 2017.

[CH13] Erika Check Hayden. Geneticists push for global data-sharing. *Nature News*, 498(7452):16, 2013.

[CLCY12] Li-Yeh Chuang, Ming-Cheng Lin, Hsueh-Wei Chang, and Cheng-Hong Yang. Odds ratio-based genetic algorithm for prediction of snp-snp interactions in breast cancer association study. In *Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International Conference on*, pages 920–925. IEEE, 2012.

[Cor02] Heather J Cordell. Epistasis: what it means, what it doesn't mean, and statistical methods to detect it in humans. *Human molecular genetics*, 11(20):2463–2468, 2002.

[Cor09] Heather J Cordell. Detecting gene–gene interactions that underlie human diseases. *Nature Reviews Genetics*, 10(6):392, 2009.

[CW97] Andrew G Clark and Lei Wang. Epistasis in measured genotypes: Drosophila p-element insertions. *Genetics*, 147(1):157–163, 1997.

[DMN18] Anindya De, Elchanan Mossel, and Joe Neeman. Is your data low-dimensional? *arXiv preprint arXiv:1806.10057*, 2018.

[DUDA06] Ramón Díaz-Uriarte and Sara Alvarez De Andres. Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 7(1):3, 2006.

[Fel12] Vitaly Feldman. A complete characterization of statistical query learning with applications to evolvability. *Journal of Computer and System Sciences*, 78(5):1444–1459, 2012.

[FGKP06] Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. New results for learning noisy parities and halfspaces. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 563–574. IEEE, 2006.

[Fis19] Ronald A Fisher. Xv.—the correlation between relatives on the supposition of mendelian inheritance. *Earth and Environmental Science Transactions of the Royal Society of Edinburgh*, 52(2):399–433, 1919.

[FMI+06] Marc PC Fossorier, Miodrag J Mihaljević, Hideki Imai, Yang Cui, and Kanta Matsuura. An algorithm for solving the lpn problem and its application to security evaluation of the hb protocols for rfid authentication. In *International Conference on Cryptology in India*, pages 48–62. Springer, 2006.

[GJL14] Qian Guo, Thomas Johansson, and Carl Löndahl. Solving lpn using covering codes. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 1–20. Springer, 2014.

[GPDHGPR13] NicoláS GarcíA-Pedrajas, Aida De Haro-GarcíA, and Javier PéRez-RodríGuez. A scalable approach to simultaneous evolutionary instance and feature selection. *Information Sciences*, 228:150–174, 2013.

[GS95] Sally A. Goldman and Robert H. Sloan. Can pac learning algorithms tolerate random attribute noise? *Algorithmica*, 14(1):70–84, 1995.

[GT92] Sun Wei Guo and Elizabeth A Thompson. Performing the exact test of hardy-weinberg proportion for multiple alleles. *Biometrics*, pages 361–372, 1992.

[GWBV02] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3):389–422, 2002.

[Hal99] Mark Andrew Hall. Correlation-based feature selection for machine learning. 1999.

[HC97] DL Hartl and Ar G Clark. Principles of population genetics sinauer associates. *Sunderland, MA*, 1997.

[HG15] Zena M Hira and Duncan F Gillies. A review of feature selection and feature extraction methods applied on microarray data. *Advances in bioinformatics*, 2015, 2015.

[HHE04] Estevam R Hruschka, Eduardo R Hruschka, and Nelson FF Ebecken. Feature selection by bayesian networks. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 370–379. Springer, 2004.

[Hua15] Samuel H Huang. Supervised feature selection: A tutorial. *Artificial Intelligence Research*, 4(2):22, 2015.

[HY08] Ingileif B Hallgrímsdóttir and Debbie S Yuster. A complete classification of epistatic two-locus models. *BMC genetics*, 9(1):17, 2008.

[İT13a] İlhan İlhan and Gülay Tezel. A genetic algorithm–support vector machine method with parameter optimization for selecting the tag snps. *Journal of biomedical informatics*, 46(2):328–340, 2013.

[İT13b] İlhan İlhan and Gülay Tezel. How to select tag snps in genetic association studies? the clontagger method with parameter optimization. *Omics: a journal of integrative biology*, 17(7):368–383, 2013.

[JA06] Peyman Jafari and Francisco Azuaje. An assessment of recently published gene expression data analyses: reporting experimental design and statistical factors. *BMC Medical Informatics and Decision Making*, 6(1):27, 2006.

[JDC+04] Hongying Jiang, Youping Deng, Huann-Sheng Chen, Lin Tao, Qiuying Sha, Jun Chen, Chung-Jui Tsai, and Shuanglin Zhang. Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes. *BMC bioinformatics*, 5(1):81, 2004.

[JUA05] Thanyaluk Jirapech-Umpai and Stuart Aitken. Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes. *BMC bioinformatics*, 6(1):148, 2005.

[Kat07] Jonathan Katz. Efficient cryptographic protocols based on the hardness of learning parity with noise. In *IMA International Conference on Cryptography and Coding*, pages 1–15. Springer, 2007.

[KC13] V Anuja Kumari and R Chitra. Classification of diabetes disease using support vector machine. *International Journal of Engineering Research and Applications*, 3(2):1797–1801, 2013.

[Kea98] Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.

[KKHVA17] C Kotsavasiloglou, N Kostikis, Dimitrios Hristu-Varsakelis, and M Arnaoutoglou. Machine learning-based classification of simple drawing movements in parkinson's disease. *Biomedical Signal Processing and Control*, 31:174–180, 2017.

[KLN+11] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

[KMV08] Adam Tauman Kalai, Yishay Mansour, and Elad Verbin. On agnostic boosting and parity learning. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 629–638. ACM, 2008.

[Kno14] Bartha Maria Knoppers. Framework for responsible sharing of genomic and health-related data. *The HUGO journal*, 8(1):3, 2014.

[KPJM12] Asha Gowda Karegowda, V Punya, MA Jayaram, and AS Manjunath. Rule based classification for diabetic patients using cascaded k-means and decision tree c4. 5. *International Journal of Computer Applications*, 45(12):45–50, 2012.

[KTS+17] Ioannis Kavakiotis, Olga Tsave, Athanasios Salifoglou, Nicos Maglaveras, Ioannis Vlahavas, and Ioanna Chouvarda. Machine learning and data mining methods in diabetes research. *Computational and structural biotechnology journal*, 15:104–116, 2017.

[LF06] Éric Levieil and Pierre-Alain Fouque. An improved lpn algorithm. In *International Conference on Security and Cryptography for Networks*, pages 348–359. Springer, 2006.

[LGR+07] Nanye Long, Daniel Gianola, Guilherme JM Rosa, KA Weigel, and S Avendano. Machine learning classification procedure for selecting snps in genomic selection: application to early mortality in broilers. *Journal of animal breeding and genetics*, 124(6):377–389, 2007.

[LGW+14] Pei Li, Maozu Guo, Chunyu Wang, Xiaoyan Liu, and Quan Zou. An overview of snp interactions in genome-wide association studies. *Briefings in functional genomics*, 14(2):143–155, 2014.

[LMN93] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, fourier transform, and learnability. *Journal of the ACM (JACM)*, 40(3):607–620, 1993.

[LPH+13] Hsi-Che Liu, Pei-Chen Peng, Tzung-Chien Hsieh, Ting-Chi Yeh, Chih-Jen Lin, Chien-Yu Chen, Jen-Yin Hou, Lee-Yung Shih, and Der-Cherng Liang. Comparison of feature selection methods for cross-laboratory microarray analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 10(3):593–604, 2013.

[Lyu05] Vadim Lyubashevsky. The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem. In *Approximation, randomization and combinatorial optimization. Algorithms and techniques*, pages 378–389. Springer, 2005.

[MG13] Pradipta Maji and Partha Garai. On fuzzy-rough attribute selection: criteria of max-dependency, max-relevance, min-redundancy, and max-significance. *Applied Soft Computing*, 13(9):3968–3980, 2013.

[MM09] Weidong Mao and Jinghe Mao. The application of apriori-gen algorithm in the association study in type 2 diabetes. In *Bioinformatics and Biomedical Engineering, 2009. ICBBE 2009. 3rd International Conference on*, pages 1–4. IEEE, 2009.

[MP69] Marvin Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry*. Cambridge, MA, USA, 1969.

[MRRM06] Brett A McKinney, David M Reif, Marylyn D Ritchie, and Jason H Moore. Machine learning for detecting gene-gene interactions. *Applied bioinformatics*, 5(2):77–88, 2006.

[MRRR+16] Paulino Martinez, Diego Robledo, Silvia T Rodriguez-Ramilo, Miguel Hermida, Xoana Taboada, Patricia Pereiro, Juan A Rubiolo, Laia Ribas, Antonio Gomez-Tato, Jose Antonio Alvarez-Dios, et al. Turbot (scophthalmus maximus) genomic resources: application for boosting aquaculture production. In *Genomics in aquaculture*, pages 131–163. Elsevier, 2016.

[MS16] Fahimeh Mostofi and Fahreddin Sadikoglu. Discovering snp interactions associated with breast cancer using evolutionary algorithms. *Procedia Computer Science*, 102:562–569, 2016.

[MSH07] Shuangge Ma, Xiao Song, and Jian Huang. Supervised group lasso with applications to microarray data analysis. *BMC bioinformatics*, 8(1):60, 2007.

[MW07] Jason H Moore and Bill C White. Tuning relieff for genome-wide genetic analysis. In *European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, pages 166–175. Springer, 2007.

[O'D14] Ryan O'Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014.

[OT03] CH Ooi and Patrick Tan. Genetic algorithms applied to multi-class prediction for the analysis of gene expression data. *Bioinformatics*, 19(1):37–44, 2003.

[Pie12] Krzysztof Pietrzak. Cryptography from learning parity with noise. In *International Conference on Current Trends in Theory and Practice of Computer Science*, pages 99–114. Springer, 2012.

[PSGK16] Sajida Perveen, Muhammad Shahbaz, Aziz Guergachi, and Karim Keshavjee. Performance analysis of data mining classification techniques to predict diabetes. *Procedia Computer Science*, 82:115–121, 2016.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):34, 2009.

[Reg10] Oded Regev. The learning with errors problem. *Invited survey in CCC*, 7, 2010.

[Riv91] Ronald L Rivest. Cryptography and machine learning. In *International Conference on the Theory and Application of Cryptology*, pages 427–439. Springer, 1991.

[RJFD10] Andrea Rau, Florence Jaffrézic, Jean-Louis Foulley, and Rebecca W Doerge. An empirical bayesian method for estimating biological networks from temporal microarray data. *Statistical Applications in Genetics and Molecular Biology*, 9(1), 2010.

[SAK05] John D Storey, Joshua M Akey, and Leonid Kruglyak. Multiple locus linkage analysis of genomewide expression in yeast. *PLoS biology*, 3(8):e267, 2005.

[SB13] Vasin Suttichaya and Pattarasinee Bhattarakosol. Solving the learning parity with noise's open question. *Information Processing Letters*, 113(14-16):562–566, 2013.

[SB18] Mahsa Shabani and Pascal Borry. Rules for processing genetic data for research purposes in view of the new eu general data protection regulation. *European Journal of Human Genetics*, 26(2):149, 2018.

[SCY+16] Liming Shen, Huiling Chen, Zhe Yu, Wenchang Kang, Bingyu Zhang, Huaizhong Li, Bo Yang, and Dayou Liu. Evolving support vector machines using fruit fly optimization for medical data classification. *Knowledge-Based Systems*, 96:61–75, 2016.

[SI07] Holger Schwender and Katja Ickstadt. Identification of snp interactions using logic regression. *Biostatistics*, 9(1):187–198, 2007.

[Slo88] Robert Sloan. Types of noise in data for concept learning. In *Proceedings of the first annual workshop on Computational learning theory*, pages 91–96. Morgan Kaufmann Publishers Inc., 1988.

[UGEFC12] Rosanna Upstill-Goddard, Diana Eccles, Joerg Fliege, and Andrew Collins. Machine learning approaches for the discovery of gene–gene interactions in disease data. *Briefings in bioinformatics*, 14(2):251–260, 2012.

[UKSA+12] Ryan J Urbanowicz, Jeff Kiralis, Nicholas A Sinnott-Armstrong, Tamra Heberling, Jonathan M Fisher, and Jason H Moore. Gametes: a fast, direct algorithm for generating pure, strict, epistatic models with random architectures. *BioData mining*, 5(1):16, 2012.

[UMLC+18] Ryan J Urbanowicz, Melissa Meeker, William La Cava, Randal S Olson, and Jason H Moore. Relief-based feature selection: introduction and review. *Journal of biomedical informatics*, 85:189–203, 2018.

[VA15a] V Veena Vijayan and C Anjali. Computerized information system using stacked generalization for diagnosis of diabetes mellitus. In *Intelligent Computational Systems (RAICS), 2015 IEEE Recent Advances in*, pages 173–178. IEEE, 2015.

[VA15b] V Veena Vijayan and C Anjali. Prediction and diagnosis of diabetes mellitus—a machine learning approach. In *Intelligent Computational Systems (RAICS), 2015 IEEE Recent Advances in*, pages 122–127. IEEE, 2015.

[Val84] Leslie G Valiant. A theory of the learnable. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 436–445. ACM, 1984.

[Val85] Leslie G Valiant. Learning disjunction of conjunctions. In *IJCAI*, pages 560–566, 1985.

[Val12] Gregory Valiant. Finding correlations in subquadratic time, with applications to learning parities and juntas. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 11–20. IEEE, 2012.

[WWC16] Lipo Wang, Yaoli Wang, and Qing Chang. Feature selection methods for big data bioinformatics: a survey from the search perspective. *Methods*, 111:21–31, 2016.

[YLCC13] Cheng-Hong Yang, Yu-Da Lin, Li-Yeh Chaung, and Hsueh-Wei Chang. Evaluation of breast cancer susceptibility using improved genetic algorithms to generate genotype snp barcodes. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 10(2):361–371, 2013.

[YMLC16] Cheng-Hong Yang, Sin-Hua Moi, Yu-Da Lin, and Li-Yeh Chuang. Genetic algorithm combined with a local search method for identifying susceptibility genes. *Journal of Artificial Intelligence and Soft Computing Research*, 6(3):203–212, 2016.

[YWCY17] Cheng-Hong Yang, Zi-Jie Weng, Li-Yeh Chuang, and Cheng-San Yang. Identification of snp-snp interaction for chronic dialysis patients. *Computers in biology and medicine*, 83:94–101, 2017.

[YZZZ10] Pengyi Yang, Bing B Zhou, Zili Zhang, and Albert Y Zomaya. A multi-filter enhanced genetic ensemble system for gene selection and sample classification of microarray data. *BMC bioinformatics*, 11(1):S5, 2010.

[ZXX+17] Tao Zheng, Wei Xie, Liling Xu, Xiaoying He, Ya Zhang, Mingrong You, Gong Yang, and You Chen. A machine learning-based framework to identify type 2 diabetes through electronic health records. *International journal of medical informatics*, 97:120–127, 2017.