Sample size determination in multi-class classification and prediction based on single-nucleotide polymorphisms

by

Xinyu Liu

(Under the direction of T.N. Sriram)

Abstract

Single-nucleotide polymorphisms (SNPs), believed to determine human differences, are widely used to predict risk of diseases and class membership of subjects. In the literature, several supervised machine learning methods, such as, support vector machine, neural network and logistic regression, are available for classification. Typically, however, samples for training a machine are limited and/or the sampling cost is high. Thus, it is essential to determine the minimum sample size needed to construct a classifier based on SNP data. Such a classifier would facilitate correct classification while keeping the sample size to a minimum, thereby making the studies cost-effective.

In this dissertation, for coded SNP data from two classes, an optimal classifier and an approximation to its probability of correct classification (PCC) are derived. A linear classifier is constructed and an approximation to its PCC is also derived. These approximations are validated through a variety of Monte Carlo simulations. A sample size determination algorithm based on the criterion which ensures that the difference between the two approximate $PCCs$ is below a threshold, is given. For the HapMap data on Chinese and Japanese populations, a linear classifier is built using 51 independent SNPs, and the required total sample sizes are determined using our algorithm.

For coded SNP data from $D(\geq 2)$ classes, we derive an optimal Bayes classifier and a linear classifier, and obtain a normal approximation to the PCC for each classifier. These approximations are used to evaluate the associated Area Under the Receiver Operating Characteristic (ROC) Curve ($AUCs$) or Volume Under the ROC hyper-Surface ($VUSs$). We give an algorithm for sample size determination, which ensures that the difference between the two approximate $AUCs$ (or $VUSs$) is below a pre-specified threshold. The performance of this algorithm is also illustrated via simulations. For the *HapMap* data with three and four populations, a linear classifier is built using 92 independent SNPs and the required total sample sizes are determined. We also illustrate the usefulness of our sample size determination algorithm in a prediction problem using a *Heterogeneous Stock Mice* data.

INDEX WORDS: Area Under the Receiver Operating Characteristic Curve; Classification; Hapmap data; Heterogeneous Stock Mice data; Probability of correct classification; Receiver Operating Characteristic; Sample Size Determination; Single-nucleotide polymorphisms; Volume Under the Receiver Operating Characteristic hyper-Surface; Wald test.

Sample size determination in multi-class classification and prediction based on single-nucleotide polymorphisms

by

Xinyu Liu

B.S., Fudan University, 2004

M.S., University of Copenhagen, 2007

M.S., University of Georgia, 2010

A Dissertation Submitted to the Graduate Faculty

of The University of Georgia in Partial Fulfillment

of the

Requirements for the Degree

Doctor of Philosophy

Athens, Georgia

2013

Sample size determination in multi-class classification and prediction based on single-nucleotide polymorphisms

by

Xinyu Liu

Approved:

Major Professor: T.N. Sriram

Committee: William McCormick
Jaxk Reeves
Lily Wang
Xiangrong Yin

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
August 2013

Dedication

To my parents, and my wife Liang.

Acknowledgments

First of all, I would like to express my sincere appreciation to my advisor, Professor T. N. Sriram, for all of his advice, guidance, and encouragement. He has demonstrated that he is a masterly statistician, an amazing teacher, and a great communicator during my research years.

I also want to thank the other members of my committee: Dr. Jaxk Reeves, Dr. Lilly Wang, Dr. William P. McCormick, and Dr. Xiangrong Yin for their direct and fundamental contributions to this dissertation.

Finally and most importantly, a special thanks is extended to my parents and my wife, Liang. Without their invaluable love and constant support, I could never have accomplished the most important goal in my life.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction and Literature Review

1.1 Single-Nucleotide Polymorphisms and a model

Humans usually have 23 pairs of *chromosomes* and this entire set is called a *genome*. Each human being shares 99.5% of the Deoxyribonucleic acid (DNA) in their chromosomes with other humans. Therefore, the question is: *What makes us different from one another*? The answer lies in what are called *Single Nucleotide Polymorphisms* (SNPs), pronounced *snips*.

The human genome is composed of 3 billion base pairs of single nucleotides, **A**, **T**, **C** and **G**. To make new cells, an existing cell divides in two. But first it copies its DNA so the new cells will each have a complete set of genetic instructions. Usually, the body does not make many mistakes in the copying process. However, no system is perfect. Sometimes, when a genome is copied to make a new cell, a single base pair is left out, added or substituted. Variation at a single base pair is called the SNP. For example, a SNP may replace the nucleotide guanosine (**G**) with the nucleotide thymine (**T**) in a certain stretch of DNA, as shown in the 5th locus of Figure 1.1. In this case, we say that there are two alleles, **G** and **T**, and almost all common SNPs have only two alleles; `https://www.23andme.com/gen101/snps/`

SNPs occur normally throughout a person's DNA. They occur once in every 300 nucleotides on the average, which means that there are roughly 10 million SNPs in the human genome and these account for many of the genetic differences between humans. SNPs can act as biological markers, helping scientists locate genes that are associated with disease. When SNPs occur within a gene or in a regulatory region near a gene, they may play a more direct role in disease by affecting the gene's function. Most SNPs have no

effect on health or development. Some of these genetic differences, however, have proven to be very important in the study of human health. Researchers have found SNPs that may help predict an individual's response to certain drugs, susceptibility to environmental factors such as toxins, and risk of developing particular diseases. SNPs can also be used to track the inheritance of disease genes within families. Future studies will work to identify SNPs associated with complex diseases such as heart disease, diabetes, and cancer; see `http://ghr.nlm.nih.gov/handbook/genomicresearch/snp`.

In each position of base pair (locus), there are often two types of alleles measured by the frequency of occurrence, called the Major allele ("a") and the Minor allele ("A"). According to the definition of SNP, the Minor allele should occur more than 1%, or else it is called mutation; see Figure 1.1. SNPs fall in coding region (may affect protein expression), non-coding region (affects gene splicing, transcription factor binding, or sequence of non-coding RNA), or even in non-gene region. Most often, researchers are mainly interested in the SNP from gene region.

In this dissertation, we use the Hardy-Weinberg equilibrium model to study SNPs (Crow,1999; Emigh, 1980). According to this model, the allele frequency in a population is constant, that is, the frequency of "A" is θ and the frequency of "a" is $(1-\theta)$. The Hardy Weinberg equilibrium model assumes that the frequency of "AA" is θ^2, the frequency of "Aa" is $2\theta(1-\theta)$ and the frequency of "aa" is $(1-\theta)^2$. It is common to convert the genotype ("aa", "Aa" and "AA") into number of minor alleles. That is, we code the jth SNP by the number $X_j(=0,1,2)$, which denotes the number of minor alleles in the genotype "aa", "Aa" and "AA", respectively. This framework is valid because, for the SNPs in coding region, the number of minor alleles decide the amount of protein expressed in that region.

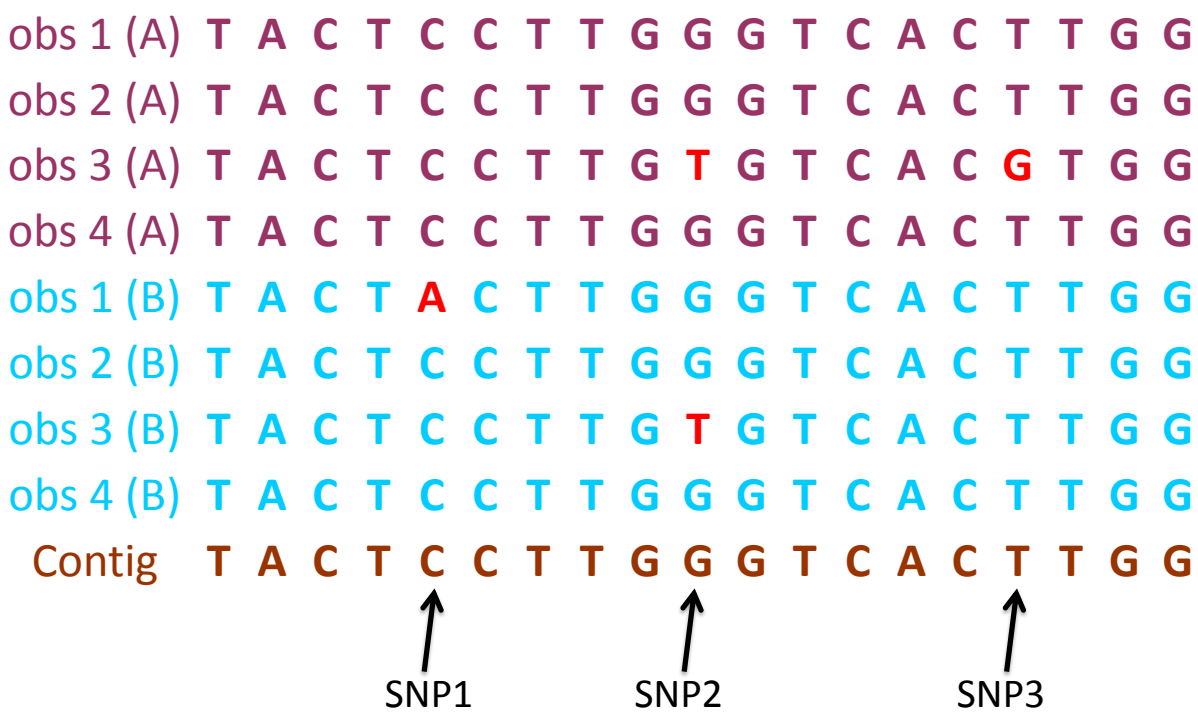

Figure 1.1: Description of SNP. If the sequences of population from one species are put together, and made multiple alignment, then the bases in the same position from whole population lie in one column. For most of columns, they include the same base (allele), but for a few columns (e.g. in locus 5, 10 and 16), they include different allele, which is called SNP.

1.2 Classifier and Predictor Based on Single-Nucleotide Polymorphisms

One of the main aims of this dissertation is to construct a classifier based on the coded SNP data, $\{X_j\}$ defined in Section 1.1. By a classifier, we mean a statistical mechanism that classifies a new observation to an appropriate sub-population. Usually, researchers use training datasets to train a classifier, and use a validating dataset to test the goodness of the trained classifier. There are many classifiers available in the literature such as *Fisher's Linear Discriminant function, Logistic Regression, genetic programming, Support Vector Machine* (SVM), *Neural Network* (NN) and *Decision Tree*, to name a few. Next, we give briefly describe two popular classifiers– SVM and NN.

The Support Vector Machine (SVM) classifies using an N-dimensional hyperplane (with a sigmoid kernel function) that optimally separates the data into two classes. A good separation is achieved by the hyperplane that has the largest distance to the nearest training data points of any class. If the dataset cannot be separated linearly, then it can be mapped into a higher dimensional space; ideally this will make the separation easier in that space. For more details on research using SVM, see Guzzetta, Jurman and Furlanello, 2010; Kuznetsov, McDuffie and Moslehi, 2009; Onuki, Shibuya and Kanehisa, 2010; and Davies *and others*, 2010.

Neural Network (NN) is inspired by the biological neurons, which are usually made of input layer neurons, hidden layer neurons and output layer neurons. Usually, each neuron in the input layer corresponds to one predictor, the one in the hidden layer corresponds to a linear combination of the neurons in input layer, and the output layer corresponds to linear combinations of the neurons in hidden layer. The neuron corresponds to one value, and the line between two neurons is the weight in the linear combination. For more details on research using NN, see Sabbagh and Darlu, 2006; Wang and Larder, 2003. For more details on genetic programming, see Nunkesser *and others* (2007) and for details on Logistic Regression, see Davies *and others* (2010).

Many researchers have shown recently that a large number of SNPs affect the genetic architecture of complex traits; see, for instance, Willer *and others* (2008); Sanna *and others*

(2008); Harley *and others* (2009); Zanke *and others* (2007); Yeager *and others* (2007); WTCCC (2007); Winkelmann *and others* (2007); Weedon *and others* (2007); Saldek *and others* (2007); Scuteri *and others* (2007); Scott *and others* (2007); Saxena *and others* (2007); Rioux *and others* (2007); Moffatt *and others* (2007); and Libioulle *and others* (2007). SNP data have also been widely used in predicting the phenotypes such as ethnicity, quantitative traits, or risk of diseases (Guzzetta, Jurman and Furlanello, 2010; Lee *and others* (2008); Nunkesser *and others*, 2007; Wary, Goddard and Visscher, 2007; Zhou and Wang, 2007).

Whereas classification of an observation into one of D classes based on an SNP data is important, predicting the value of a future phenotype based on SNP data are also of interest. Lee *and others* (2008) predicted three traits of stock mouse, including coat color (measure of darkness), percentage of $CD8+$ cells and mean cellular haemoglobin, using an additive genetic model. Wray *and others* (2007) built a model to predict the risk of a disease. Wei *and others* (2009) predicted the risk of type 1 diabetes using the support vector machine. Lorenzana (2009) predicted plant traits, such as, plant height, ear height, root lodging, and so on, using multiple linear regression and best linear unbiased prediction. De Roos *and others* (2009) built a model to predict the breeding value of bulls.

Despite the significance of SNPs, the question that has not been addressed yet is : *What is the sample size required to build an accurate predictor of class membership based on coded SNP data*? This dissertation focuses squarely on answering this question. Theoretically, a larger sample size leads to a higher prediction accuracy, but in reality, clinical samples are often limited and/or the cost of sampling is high. In this dissertation, we develop an algorithm to determine a (total) training sample size that is just large enough to satisfy a pre-specified accuracy of a linear classifier based on SNPs. Before we propose our sample size determination method based on SNPs, we will give a brief review of sample determination methods based on microarray data.

1.3 Review of Sample Size Determination Based on Microarray Data

In the literature, there are a variety of sample size determination methods for classification based on microarrays; see, for instance, De Valpine *and others* (2009); Dobbin and Simon (2005, 2007); and Dobbin, Zhao and Simon (2008). In a microarray data, typically the number of predictors exceeds the number of observations. Therefore, most classification methods based on microarrays are divided into two steps– the first step selects the significant variables and the second step builds the classification model using the selected variables.

Mukherjee *and others* (2003) introduced a sample size determination method for classifying microarray data using *learning curves*. This method samples a training data from the original dataset, computes the error rate from the model built by the sampled training data, and then fits the learning curve by the error rate and sample size. The model $e(n) = an^{-\alpha} + b$ is used in fitting, where $e(n)$ is the expected error rate with sample size n, a is learning rate, α is the decay rate, and b is the Bayes error. To obtain the learning curve, a, b and α are estimated by the data points with different sample size, together with their corresponding error rate. This method can be used in all models, and in Mukherjee's paper, it is applied in cancer non-cancer classification by Microarray dataset, and the methodology gives reasonable estimates, but it needs relative large dataset available to estimate the learning curve.

Fu *and others* (2005) assumed that a classifier is developed sequentially (one-at-a-time) and provided a stopping criteria which stops sampling when the probability of misclassifying a new subject is below a threshold. They introduced a formula for stopping rule using the Martingale Central Limit Theorem (CLT). Their sample size determination method updates a classifier sequentially and thus avoids the potential problem of over sampling. Moreover, their method is not classifier-specific and hence it can be used with any classifier such as the SVM, Classification and Regression Trees (CART) etc. Fu *and others* (2005) applied their method to microarray datasets, including breast cancer data and breast tumor characterization data, using the Linear Discriminant Analysis (LDA) and K-Nearest Neighbors

(KNN) classifiers. A drawback of this approach is that it does not provide an estimate of the expected number of cases needed.

Dobbin and Simon (2007) recently considered a two-class problem based on gene expression data and introduced a classifier-specific method to estimate the sample size needed in classification. More specifically, they assumed that the gene expression data vector from each class follows a multivariate normal distribution, where the covariance matrices are the same but the mean vectors are different. Using the normality assumption and ideas from Fisher's discriminant function, they introduced an optimal classifier and obtained an upper bound for the theoretical probability of correct classification (PCC), denoted by $PCC(\infty)$. They also introduced a linear classifier and computed an expected PCC with sample size n, denoted by $PCC(n)$. For each threshold (γ), they introduced a criterion which determines the sample size needed to satisfy $PCC(\infty) - PCC(n) < \gamma$. They illustrated their sample size determination method using simulations and data analysis.

Note, however, that the sample size determination methods mentioned above assume normality and currently, there no sample size determination methods when the data is discrete. As mentioned earlier, this dissertation focuses on the problem of sample size determination for multi-class classification, when the observations are coded SNPs.

1.4 Outline of the Dissertation

In Chapter 2, we consider the two-class scenario and develop a sample size determination method for an SNP based classifier. More specifically, for coded SNP data from two classes, an optimal classifier and an approximation to its PCC are derived. A linear classifier is constructed and an approximation to its PCC is also derived. These approximations are validated through a variety of Monte Carlo simulations. A sample size determination algorithm based on the criterion which ensures that the difference between the two approximate *PCCs* is below a threshold is given and its effectiveness is illustrated via simulations. The

method is validated by Monte Carlo simulation, and applied into Hapmap data for two-class classification. See chapter 2 for details.

In Chapter 3, we generalize our sample size determination method for a two-class scenario to a multi-class scenario. Once again, we introduce a SNP based classifier, but the performance of the classifier is evaluated based on the area under the receiver operating characteristic (ROC) curve (AUC) for a two-class classifier and volume under the ROC hypersurface (VUS) for multi-class classifier. For coded SNP data, an optimal classifier and an approximation to its AUC/VUS, together with a linear classifier and an approximation to its AUC/VUS, are derived. These approximations are validated through a series of Monte Carlo simulations. A sample size determination algorithm based on the criterion, which ensures that the difference between the two approximate AUC/VUS is below a threshold is given and its effectiveness is illustrated via simulations. Our method is applied to Hapmap data with 3 and 4 populations. This method is also generalized to an SNP predictor, and applied to the Heterogeneous Stock Mice Data.

Note that each chapter is self-contained in terms of development and assessment of the above methods, but we give an overall conclusion for both the chapters in Chapter four.

1.5 References

[1] Crow,J.F. (1999). Hardy, Weinberg and language impediments. *Genetics*, **152**, 821-825.

[2] Davies,R.W., Dandona,S., Stewart,A.F.R., Chen,L., Ellis,S.G., Tang,W.H.W., Hazen,S.L., Roberts,R., McPherson,R., and Wells,G.A. (2010) Improved prediction of cardiovascular disease based on a panel of single nucleotide polymorphisms identified through genome-wide association studies. *Circ Cardiovasc Genet* **3**, 468-474.

[3] De Roos,A.P.W., Hayes,B.J. and Goddard,M.E. (2009) Reliability of Genomic Predictions Across Multiple Populations. *Genetics*, **183**, 1545-1553.

[4] De Valpine,P., Bitter,H.M., Brown,M.P.S., and Heller,J. (2009) A simulation-approximation approach to sample size planning for high-dimensional classification studies. *Biostatistics*, **10**, 424-435.

[5] Dobbin,K.K. and Simon,R.M. (2005) Sample size determination in microarray experiments for class comparison and prognostic classification. *Biostatistics*, **6**, 27-38.

[6] Dobbin,K.K. and Simon,R.M. (2007) Sample size planning for developing classifiers using high-dimensional DNA microarray data. *Biostatistics*, **8**, 101-117.

[7] Dobbin,K.K., Zhao,Y., and Simon,R.M. (2008) How large a training set is needed to develop a classifier for microarray data. *Clin Cancer Res*, **14**, 108-114.

[8] Emigh, T.H. (1980). A Comparison of Tests for Hardy-Weinberg Equilibrium. *Biometrics*, **36**, 627-642.

[9] Fu,W.J., Dougherty,E.R., Mallick,B., and Carroll,R.J. (2005) How many samples are needed to build a classifier: a general sequential approach. *Bioinformatics*, **21**, 63-70.

[10] Guzzetta,G., Jurman,G., and Furlanello,C. (2010). A machine learning pipeline for quantitative phenotype prediction from genotype data. *BMC Bioinformatics*, **11**(Suppl 8), S3.

[11] Harley,J.B., Alarcon-Riquelme,M.E., Criswell,L.A., Jacob,C.O., Kimberly,R.P., et al. (2008) Genome-wide association scan in women with systemic lupus erythematosus identifies susceptibility variants in ITGAM, PXK, KIAA1542 and other loci. *Nat Genet*, **40**, 204-210.

[12] Kuznetsov,I.B., McDuffie,M., and Moslehi,R. (2009) A web server for inferring the human N-acetyltransferase-2 (NAT2) enzymatic phenotype from NAT2 genotype. *Bioinformatics*, **25**, 1185-1186.

[13] Lachenbruch,P.A. (1968). On expected probabilities of misclassification in discriminant analysis, necessary sample size, and a relation with the multiple correlation coefficient. *Biometrics*, **24**, 823-834.

[14] Lee,S.H., van der Werf,J.H.J., Hayes,B.J., Goddard,M.E., and Visscher,P.M. (2008) Predicting Unobserved Phenotypes for Complex Traits from Whole-Genome SNP Data. *Plos Genet*, **4**, e1000231.

[15] Libioulle,C., Louis,E., Hansoul,S., Sandor,C., Farnir,F., et al. (2007) Novel Crohn disease locus identified by genome-wide association maps to a gene desert on 5p13.1 and modulates expression of PTGER4. *PLoS Genet.*, **3**, e58.

[16] Lorenzana,R.E., Bernardo,R. (2009) Accuracy of genotypic value predictions for marker-based selection in biparental plant populations. *Theor. Appl. Genet.*, **120**, 151-161.

[17] Moffatt,M.F., Kabesch,M., Liang,L., Dixon,A.L., Strachan,D., et al. (2007) Genetic variants regulating ORMDL3 expression contribute to the risk of childhood asthma. *Nature*, **448**, 470-473.

[18] Mukherjee,S., Tamayo,P., Rogers,S., Rifkin,R., Engle,A., Campbell,C., Golub,T.R., and Mesirov,J.P. (2003) Estimating dataset size requirements for classifying DNA microarray data. *J Comput Biol*, **10**, 119-142.

[19] Nunkesser,R., Bernholt,T., Schwender,H., Ickstadt,K., and Wegener,I. (2007) Detecting high-order interactions of single nucleotide polymorphisms using genetic programming. *Bioinformatics*, **23**, 3280-3288.

[20] Onuki,R., Shibuya,T., and Kanehisa,M. (2010) New kernel methods for phenotype prediction from genotype data. *Genome Inform*, **22**, 132-141.

[21] Rioux,J.D., Xavier,R.J., Taylor,K.D., Silverberg,M.S., Goyette,P., et al. (2007) Genome-wide association study identifies new susceptibility loci for Crohn disease and implicates autophagy in disease pathogenesis. *Nat. Genet.*, **39**, 596-604.

[22] Sabbagh,A. and Darlu,P. (2006) SNP selection at the NAT2 locus for an accurate prediction of the acetylation phenotype. *Genet Med*, **8**, 76-85.

[23] Sanna,S., Jackson,A.U., Nagaraja,R., Willer,C.J., Chen,W.M., et al. (2008) Common variants in the GDF5-UQCC region are associated with variation in human height. *Nat. Genet.*, **40**, 198-203.

[24] Saxena,R., Voight,B.F., Lyssenko,V., Burtt,N.P., de Bakker,P.I., et al. (2007) Genome-wide association analysis identifies loci for type 2 diabetes and triglyceride levels. *Science*, **316**, 1331-1336.

[25] Scott,L.J., Mohlke,K.L., Bonnycastle,L.L., Willer,C.J., Li,Y., et al. (2007) A genome-wide association study of type 2 diabetes in Finns detects multiple susceptibility variants. *Science*, **316**, 1341-1345.

[26] Scuteri,A., Sanna,S., Chen,W.M., Uda,M., Albai,G., et al. (2007) Genome-Wide Association Scan Shows Genetic Variants in the FTO Gene Are Associated with Obesity-Related Traits. *PLoS Genet.*, **3**, e115.

[27] Sladek,R., Rocheleau,G., Rung,J., Dina,C., Shen,L., et al. (2007) A genome-wide association study identifies novel risk loci for type 2 diabetes. *Nature*, **445**, 881-885.

[28] Wang,D.C. and Larder,B. (2003) Enhanced prediction of lopinavir resistance from genotype by use of artificial neural networks. *J Infect Dis*, **188**, 653-660.

[29] Weedon,M.N., Lettre,G., Freathy,R.M., Lindgren,C.M., Voight,B.F., et al. (2007) A common variant of HMGA2 is associated with adult and childhood height in the general population. *Nat. Genet.*, **39**, 1245-1250.

[30] Wei,Z., Wang,K., Qu,H.Q., Zhang,H., Bradfield,J., et al. (2009) From Disease Association to Risk Assessment: An Optimistic View from Genome-Wide Association Studies on Type 1 Diabetes. *PLoS Genet.*, **5**, e1000678.

[31] Willer,C.J., Sanna,S., Jackson,A.U., Scuteri,A., Bonnycastle,L.L., et al. (2008) Newly identified loci that influence lipid concentrations and risk of coronary artery disease. *Nat. Genet.*, **40**, 161-169.

[32] Winkelmann, J., Schormair, B., Lichtner, P., Ripke, S., Xiong, L., et al. (2007) Genome-wide association study of restless legs syndrome identifies common variants in three genomic regions. *Nat. Genet.*, **39** 1000-1006.

[33] Wray,N.R., Goddard,M.E., and Visscher.P.M. (2007) Prediction of individual genetic risk to disease from genome-wide association studies. *Genome Res.*, **17**, 1520-1528.

[34] WTCCC(2007) Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature*, **447**, 661-678.

[35] Yeager,M., Orr,N., Hayes,R.B., Jacobs,K.B., Kraft,P., et al. (2007) Genome-wide association study of prostate cancer identifies a second risk locus at 8q24. *Nat. Genet.*, **39**, 645-649.

[36] Zanke,B.W., Greenwood,C.M., Rangrej,J., Kustra,R., Tenesa,A., et al. (2007) Genome-wide association scan identifies a colorectal cancer susceptibility locus on chromosome 8q24. *Nat. Genet.*, **39**, 989-994.

[37] Zhou,N. and Wang,L. (2007) Effective selection of informative SNPs and classification on the HapMap genotype data. *BMC Bioinformatics*, **8**, 484.

Chapter 2

Sample size determination for classifiers based on single-nucleotide polymorphisms[1]

[1]Liu, X., Wang, Y., Rekaya, R. and Sriram, T. N. Published in: *Biostatistics,* (2012),**13**, 217-227. Reprinted here with permission of publisher.

Abstract

Single-nucleotide polymorphisms (SNPs), believed to determine human differences, are widely used to predict risk of diseases. Typically, clinical samples are limited and/or the sampling cost is high. Thus, it is essential to determine an adequate sample size needed to build a classifier based on SNPs. Such a classifier would facilitate correct classifications, while keeping the sample size to a minimum, thereby making the studies cost-effective. For coded SNP data from two classes, an optimal classifier and an approximation to its probability of correct classification (PCC) are derived. A linear classifier is constructed and an approximation to its PCC is also derived. These approximations are validated through a variety of Monte Carlo simulations. A sample size determination algorithm based on the criterion which ensures that the difference between the two approximate $PCCs$ is below a threshold, is given and its effectiveness is illustrated via simulations. For the HapMap data on Chinese and Japanese populations, a linear classifier is built using 51 independent SNPs, and the required total sample sizes are determined using our algorithm, as the threshold varies. For example, when the threshold value is 0.05, our algorithm determines a total sample size of 166 (83 for Chinese and 83 for Japanese) that satisfies the criterion.
Key words and Phrases: Classification; Hapmap data; Probability of correct classification; Sample Size Determination; Single-nucleotide polymorphisms; Wald test.

2.1 Introduction

SNP data have been widely used in predicting the phenotypes such as ethnicity, quantitative traits, or risk of diseases (Guzzetta, Jurman and Furlanello, 2010; Lee *and others*, 2008; Nunkesser *and others*, 2007; Wary, Goddard and Visscher, 2007; Zhou and Wang, 2007). While a variety of population classifiers are available in the literature, e.g., Support Vector Machine (SVM) (Guzzetta, Jurman and Furlanello, 2010; Kuznetsov, McDuffie and Moslehi, 2009; Onuki, Shibuya and Kanehisa, 2010; Davies *and others*, 2010), genetic programming (Nunkesser *and others*, 2007), Neural Network (Sabbagh and Darlu, 2006; Wang and Larder,

2003), and Logistic Regression (Davies *and others*, 2010), the question of how many samples are required to build an accurate predictor of class membership based on coded SNP data has not been addressed yet. Theoretically, a larger sample size leads to higher prediction accuracy, but in reality, clinical samples are often limited and/or the cost of sampling is high. This article presents an algorithm to determine a (total) training sample size that is just large enough to satisfy the pre-specified accuracy of a linear classifier based on SNPs.

In the literature, there are a variety of sample size determination methods for classification of microarrays; see, for instance, De Valpine *and others*, 2009; Dobbin and Simon (2005, 2007); and Dobbin, Zhao and Simon (2008). However, these are developed for continuous data satisfying the normality assumption and hence cannot be applied to classifiers based on coded SNPs. There are also other sample size determination methods for microarray data that are not classifier-specific. These include methods based on *learning curves* due to Mukherjee *and others* (2003) and those based on sequential stopping rules proposed by Fu *and others* (2005); see Dobbin and Simon (2007) for a detailed account on all these methods.

For coded SNP data from two classes, we derive an *optimal* classifier based on Bayes Law and show that the standardized optimal classifier is asymptotically normal. We also construct a linear classifier and establish the asymptotic normality of the standardized *linear* classifier. These are shown to yield approximate expressions for the PCC of the optimal classifier, $PCC(\infty)$, and that of the linear classifier, $PCC(n)$. We then adopt the objective proposed in Dobbin and Simon (Dobbin and simon, 2007) [also see Lachenbruch (1968)], to determine the total sample size such that $PCC(\infty) - PCC(n) < \gamma$, for some threshold $\gamma \in (0, 1)$. A sample size determination algorithm involving the difference between approximate expressions for $PCC(\infty)$ and $PCC(n)$ is presented. While Monte Carlo simulations corroborate our theory, the sample size determination algorithm for the HapMap data on Chinese and Japanese populations based on a linear classifier built using 51 independent SNPs illustrates the usefulness of our methodology in applications.

This chapter is organized as follows. Section 2.2 introduces the theoretical underpinnings, whereas the proofs are deferred to Appendices given under Section 2.5. Numerical illustrations, including the HapMap data analysis are given in Section 2.3. Finally, Section 2.4 gives a discussion that summarizes our findings and suggests future research directions. We begin with basic notations and assumptions.

2.2 The Method

2.2.1 *Assumptions and the postulated model*

In population classification scenarios, subjects belong to k distinct classes, e.g., different disease groups or outcome groups or case-control groups or ethnic groups. For simplicity, we suppose that there are two classes ($k = 2$) consisting of n_1 and n_2 subjects, respectively, and refer to these classes as control group (C_1) and case group (C_2) . For each subject in a class, we observe a p-dimensional SNP vector (typically $p \gg n_1$ and n_2), denoted by $\vec{x} = (x_1, \ldots, x_p)'$, where each SNP is coded according to the number of minor alleles. That is, we code the jth SNP by the number x_j $(= 0, 1, 2)$, which denotes the number of minor alleles in the genotype "aa", "Aa" and "AA", respectively.

It is possible that some of the SNPs are highly correlated. In such a case, we choose one SNP to represent highly correlated ones. To build classifiers and determine a training sample size, we make the following assumptions:

(1) The data vector, $\vec{x} = (x_1, \ldots, x_m)'$, consists only of m ($\gg n_1$ and n_2) statistically *independent* SNPs; the rest of the $(p - m)$ SNPs are not used for classification.

(2) For each $k = 1, 2$ and $j = 1, \ldots, m$, we postulate the Hardy-Weinberg equilibrium, according to which the probability mass function of the coded SNP (X_j) belonging to class k is given by

$$P_k(X_j = x_j|\theta_{k,j}) = \binom{2}{x_j}\theta_{k,j}^{x_j}(1 - \theta_{k,j})^{2-x_j}, \quad x_j = 0, 1, 2,$$

where, by the definition of SNP, $\theta_{k,j} \in (0.01, 0.5)$. Here, $\theta_{k,j} < 0.5$ because it is the minor allele frequency and $\theta_{k,j} > 0.01$ ensures that the polymorphism is not a *mutation*; see `http://en.wikipedia.org/wiki/SNP_genotyping`. Let $\vec{\theta_1} = (\theta_{1,1}, \ldots, \theta_{1,m})'$ and $\vec{\theta_2} = (\theta_{2,1}, \ldots, \theta_{2,m})'$ denote parameter vectors corresponding to classes C_1 and C_2, respectively.

(3) There is a percentage, ρ, of the chosen m independent SNPs with marginal effect on case-control. Thus, there are $\lfloor m\rho \rfloor = l$ SNPs with marginal effect on case-control. Furthermore, assume that $\sum_{j=1}^{l} [\log(\frac{\theta_{1,j}(1-\theta_{2,j})}{\theta_{2,j}(1-\theta_{1,j})})]^2 \to \infty$, as $l \to \infty$.

2.2.2 *The optimal classifier and its PCC*

By the assumptions (1) to (3), the joint mass function of $\vec{X} = (X_1, \ldots, X_l)'$ conditional on the class $C_k, k = 1, 2$ is

$$f_k(\vec{X} = \vec{x}|\vec{\theta_k}) = \prod_{j=1}^{l} \binom{2}{x_j} \theta_{k,j}^{x_j} (1 - \theta_{k,j})^{2-x_j}. \tag{2.1}$$

For any $\vec{x}$ randomly selected from the population, define $\pi_1 = P(\vec{x} \in C_1) \in (0, 1)$. Then, $\pi_2 = P(\vec{x} \in C_2) = 1 - \pi_1$. Therefore, the posterior probability of the class C_k given $\vec{x}$ is:

$$\tau_k(\vec{\theta_k}|\vec{x}) = \frac{\pi_k f_k(\vec{x}|\vec{\theta_k})}{\pi_1 f_1(\vec{x}|\vec{\theta_1}) + \pi_2 f_2(\vec{x}|\vec{\theta_2}))}$$

for $k = 1, 2$. The Bayes classification rule classifies $\vec{x}$ to C_1 if

$$\frac{\tau_1(\vec{\theta_1}|\vec{x})}{\tau_2(\vec{\theta_2}|\vec{x})} = \frac{\pi_1}{(1 - \pi_1)} \prod_{j=1}^{l} \left\{ \left(\frac{\theta_{1,j}}{\theta_{2,j}} \right)^{x_j} \left(\frac{1 - \theta_{1,j}}{1 - \theta_{2,j}} \right)^{2-x_j} \right\} > 1.$$

As shown in Appendix.1, this leads to the *optimal* classifier:

$$\text{If } \sum_{j=1}^{l} b_j x_j > K, \text{ then classify } \vec{x} \text{ to } C_1; \text{ else, classify } \vec{x} \text{ to } C_2,$$

where $b_j = \log(\frac{\theta_{1,j}(1-\theta_{2,j})}{\theta_{2,j}(1-\theta_{1,j})})$ is the log odds-ratio for x_j, and $K = \log(\frac{(1-\pi_1)}{\pi_1}) + 2\sum_{j=1}^{l} \log(\frac{1-\theta_{2,j}}{1-\theta_{1,j}})$. Furthermore, it is shown in Appendix.2 that the *PCC* for this optimal classifier is:

$$PCC(\infty) \approx \pi_1 \Phi\left(\frac{(2\sum_{j=1}^{l} b_j \theta_{1,j} - K)}{\sqrt{2\sum_{j=1}^{l} [b_j^2 \theta_{1,j}(1 - \theta_{1,j})]}} \right) + (1 - \pi_1)\Phi\left(\frac{(K - 2\sum_{j=1}^{l} b_j \theta_{2,j})}{\sqrt{2\sum_{j=1}^{l} [b_j^2 \theta_{2,j}(1 - \theta_{2,j})]}} \right) \tag{2.2}$$

where $\approx$ denotes approximation and Φ is the standard normal cumulative distribution function.

2.2.3 *A linear classifier and its PCC*

Consider the linear classifier, $\sum_{j=1}^{m} \hat{b}_j w_{j,n} x_j$, where $\hat{b}_j = \log(\frac{\hat{\theta}_{1,j}(1-\hat{\theta}_{2,j})}{\hat{\theta}_{2,j}(1-\hat{\theta}_{1,j})})$ and $w_{j,n}$ takes values either 0 or 1, which are determined from the training set via a test of hypotheses. That is, for each $j = 1, \ldots, m$, if the SNP x_j is determined to have a marginal effect on case-control by a test of hypothesis $H_0 : \theta_{1,j} = \theta_{2,j}$ versus $H_1 : \theta_{1,j} \neq \theta_{2,j}$, then $w_{j,n} = 1$ if H_0 is rejected; else $w_{j,n} = 0$. Then, the *linear* classifier is:

$$\text{If } \sum_{j=1}^{m} \hat{b}_j w_{j,n} x_j > \tilde{K}, \text{ then classify } \vec{x} \text{ to } C_1; \text{ else, classify } \vec{x} \text{ to } C_2.$$

Note that the linear classifier defined above uses plug-in estimate, $\hat{b}_j$, of the log-odds ratio (in the optimal classifier) along with $w_{j,n} (= 0 \text{ or } 1)$, which decides whether or not the jth SNP is included in the linear classifier. When $w_{j,n} = 1$, $\hat{b}_j$ denotes the strength of discriminatory contribution of the jth SNP.

In Appendix.3, we use large sample theory to derive a Wald test of level α to test H_0 versus H_1, and an expression for the power, $1-\beta_j(n_1, n_2, h_j)$, of this test, when $\theta_{1,j} - \theta_{2,j} = h_j$. The power, $1-\beta_j(n_1, n_2, h_j)$, of the test is determined using a non-central Chi-square distribution with a non-centrality parameter, which depends on $2n = n_1 + n_2$ and h_j; see Appendix.3 for details. Henceforth, we denote the power as $1 - \beta_j(n, h_j)$. In our context, we believe that the Wald test is appropriate to test H_0 versus H_1, and we can also compute the power of the test. In other real problems, one may use a different testing procedure to select features that have a marginal effect on case-control.

Let $\tilde{\eta}_j = \tilde{\eta}_j(\alpha, \rho, n, h_j) = \rho[1-\beta_j(n, h_j)]+(1-\rho)\alpha$. As in (2.2), it is shown in Appendix.4 that

$$PCC(n) \approx \pi_1 \Phi\left(\frac{2\sum_{j=1}^{m}\theta_{1,j}\tilde{\eta}_j b_j - \tilde{K}}{\sqrt{\sum_{j=1}^{m} b_j^2[2\theta_{1,j}(1-\theta_{1,j})\tilde{\eta}_j + 4\theta_{1,j}^2\tilde{\eta}_j(1-\tilde{\eta}_j)]}}\right) + (1-\pi_1)\Phi\left(\frac{\tilde{K} - 2\sum_{j=1}^{m}\theta_{2,j}\tilde{\eta}_j b_j}{\sqrt{\sum_{j=1}^{m} b_j^2[2\theta_{2,j}(1-\theta_{2,j})\tilde{\eta}_j + 4\theta_{2,j}^2\tilde{\eta}_j(1-\tilde{\eta}_j)]}}\right). \quad (2.3)$$

Note that $PCC(n)$ depends on n because the power (hence $\tilde{\eta}_j$) depends on n.

2.2.4 *Sample size determination*

The objective is to find a training sample size, n, such that

$$PCC(\infty) - PCC(n) < \gamma, \quad (2.4)$$

where $\gamma(> 0)$ is a pre-specified threshold value. In practice, we determine n such that the difference between the corresponding approximations in (2.2) and (2.3) is less than γ. In the special case when $\vec{\theta_1} = (\theta_1, \ldots, \theta_1)'$, $\vec{\theta_2} = (\theta_2, \ldots, \theta_2)'$, and $\theta_1 > \theta_2$, note that $b_j = b = \log(\frac{\theta_1(1-\theta_2)}{\theta_2(1-\theta_1)})$, and $\tilde{\eta}_j(\alpha, \rho, n, h) = \tilde{\eta}(\alpha, \rho, n, h) = \tilde{\eta}$ (see above). Consequently, by (2.2) and (2.3)

$$PCC(\infty) - PCC(n) \approx \pi_1 \Phi\left(\frac{2\rho m b\theta_1 - K}{\sqrt{2\rho m b^2\theta_1(1-\theta_1)}}\right) + (1-\pi_1)\Phi\left(\frac{K - 2\rho m b\theta_2}{\sqrt{2\rho m b^2\theta_2(1-\theta_2)}}\right) - \pi_1 \Phi\left(\frac{2mb\theta_1\tilde{\eta} - \tilde{K}}{\sqrt{2mb^2[\theta_1(1-\theta_1)\tilde{\eta} + 2\theta_1^2\tilde{\eta}(1-\tilde{\eta})]}}\right) - (1-\pi_1)\Phi\left(\frac{\tilde{K} - 2mb\theta_2\tilde{\eta}}{\sqrt{2mb^2[\theta_2(1-\theta_2)\tilde{\eta} + 2\theta_2^2\tilde{\eta}(1-\tilde{\eta})]}}\right). \quad (2.5)$$

Remark 1: Note that $\tilde{K}$ in (2.3) is yet to be specified. Since the aim is to determine a sample size that satisfies the criterion in (2.4), it is reasonable to maximize the right side of (2.3) over $\tilde{K}$; see Appendix.5 for details. Let $\tilde{K} = \tilde{K}_0$ denote such a value. However, our simulations in Section 2.3 indicates that, in some cases, the approximate value of $PCC(n)$ for $\tilde{K}_0$ might exceed that of $PCC(\infty)$ in (2.2). To avoid this, we also maximize the approximation for $PCC(\infty)$ over K, and such a value is denoted by $K = K_0$.

2.3 Numerical Results

2.3.1 *Monte Carlo simulations*

Before delving into numerical illustrations of sample size determination, we conducted elaborate Monte Carlo simulations to verify the accuracy of the approximation for $PCC(n)$ in (2.3) and study its behavior as a function of n and other parameters. Throughout this section, we consider the special case, $\vec{\theta_1} = (\theta_1, \ldots, \theta_1)'$ and $\vec{\theta_2} = (\theta_2, \ldots, \theta_2)'$ with $\theta_1 > \theta_2$ [see (2.5)], the sample sizes $n_1 = n_2 = n$, and K and $\tilde{K}$ in (2.5) are replaced by their maximum values K_0 and $\tilde{K}_0$, respectively, as mentioned in Remark 1. Table 2.1 compares the approximate values of $PCC(n)$, denoted by $\widehat{PCC(n)}$, with the Monte Carlo based estimates, $\widehat{PCC(n)}$MC, for various specifications. To obtain $\widehat{PCC(n)}$MC values, for each specification in Table 2.1, we simulated a *training* data and a *testing* data of SNPs, each having the same sample sizes. The training data was used to build the linear classifier using the methods described in Section 2.2.3, while the testing data was used to determine the frequency of correct classification of the linear classifier. This process was repeated 200 times in order to compute the average correct classification frequency, $\widehat{PCC(n)}$MC, given in Table 2.1. It is evident from Table 2.1 that the $\widehat{Bias} = \widehat{PCC(n)}\text{MC} - \widehat{PCC(n)}$ is negligible in most cases, thereby validating the use of our approximation for $PCC(n)$. Also note that both $\widehat{PCC(n)}$MC and $\widehat{PCC(n)}$ are close to $\widehat{PCC(\infty)}$, approximate values of $PCC(\infty)$.

Table 2.1: Performance of Optimal and Linear classifiers. The values of $\widehat{PCC(n)}$ and $\widehat{PCC(n)}$MC are close to each other for various model specifications. Here, $\theta_1 = 0.3, h = \theta_1 - \theta_2$, $Size = 2n$ (n for C_1, n for C_2), m is the number of independent SNPs, $\alpha = 0.01$ is the significant level for Wald tests in Section 2.2.3, and $\rho = 1$ is the percentage of the significant SNPs.

h	m	$Size$	$\widehat{PCC(\infty)}$	$\widehat{PCC(n)}$	$\widehat{PCC(n)}$MC	$\widehat{Bias}$	SVM_MC
0.01	10	60	0.5197	0.5035	0.4995	-0.0040	0.4995
0.01	10	200	0.5197	0.5035	0.5012	-0.0023	0.5012
0.01	10	2000	0.5197	0.5042	0.5021	-0.0021	0.5022
0.01	50	60	0.5437	0.5049	0.5003	-0.0046	0.5002
0.01	50	200	0.5437	0.505	0.5042	-0.0008	0.5029
0.01	50	2000	0.5437	0.5068	0.5072	0.0004	0.5067
0.01	200	60	0.5868	0.5077	0.5072	-0.0005	0.5118
0.01	200	200	0.5868	0.508	0.503	-0.0050	0.504
0.01	200	2000	0.5868	0.512	0.5147	0.0027	0.5115
0.05	10	60	0.5997	0.5216	0.5075	-0.0141	0.5072
0.05	10	200	0.5997	0.5281	0.5216	-0.0065	0.5206
0.05	10	2000	0.5997	0.5866	0.5851	-0.0015	0.5722
0.05	50	60	0.7128	0.5336	0.528	-0.0056	0.5268
0.05	50	200	0.7128	0.5506	0.5535	0.0029	0.5495
0.05	50	2000	0.7128	0.6858	0.6913	0.0055	0.6784
0.05	200	60	0.8691	0.558	0.5533	-0.0047	0.5453
0.05	200	200	0.8691	0.5941	0.6155	0.0214	0.6043
0.05	200	2000	0.8691	0.833	0.8422	0.0092	0.8331
0.2	10	60	0.8845	0.8048	0.768	-0.0368	0.7527
0.2	10	200	0.8845	0.8836	0.8662	-0.0174	0.8552
0.2	10	2000	0.8845	0.8845	0.8785	-0.0060	0.8702
0.2	50	60	0.9959	0.9681	0.9533	-0.0148	0.9558
0.2	50	200	0.9959	0.9958	0.9927	-0.0031	0.9915
0.2	50	2000	0.9959	0.9959	0.9949	-0.0010	0.9937
0.2	200	60	1	0.9999	0.9855	-0.0144	0.998
0.2	200	200	1	1	0.9988	-0.0012	1
0.2	200	2000	1	1	0.9999	-0.0001	1

To compare the performance of our linear classifier with another classifier in the literature, such as the SVM, we also computed the $PCC(n)$ values corresponding to the SVM for the above simulation setup. We used the R software with the svm() function available in e1071 package (Dimitriadou *and others* ,2005), along with other default settings. The SVM values are given in Table 2.1 under the column SVM_MC. Note that, unlike our linear classifier, there is no approximate formula available to calculate the $PCC(n)$ for SVM. Therefore, we cannot compare $\widehat{PCC(n)}$ values for our linear classifier (or the $PCC(\infty)$ values) with $PCC(n)$ values for SVM. Table 2.1 shows that the $\widehat{PCC(n)}$MC values are essentially same as those for the SVM_MC. This says that our linear classifier is as good as or slightly better than the SVM. Note that the comparison between our classifier and the SVM is based on the simulation framework, which favors our linear classifier. For other data sets, the SVM may have a better performance in terms of $PCC(n)$.

We noted in Section 2.2.3 that the approximation to $PCC(n)$ in (2.3) depends on the power of the Wald test, which increases as n increases. Due to the complicated nature of the approximate expression in (2.3), it seems challenging to theoretically establish the monotonicity of $PCC(n)$. Hence, we study this property numerically. Table 2.2 shows that $\widehat{PCC(n)}$ is a monotonically increasing function of n, with $\widehat{PCC(\infty)}$ as its upper limit. However, the rate of increase of $\widehat{PCC(n)}$ also depends on various combinations of $h(=\theta_1-\theta_2)$, the difference between the minor allele frequencies of the two groups, and the number m of independent SNPs. In fact, the values of $\widehat{PCC(n)}$ (and $\widehat{PCC(\infty)}$) increase more rapidly as h and m increase. This is reasonable because when h is large, the two groups are distinguishable, and large m implies that data contains many significant SNPs, because $\rho = 1$ in Table 2.2. Finally, in Appendices 6, 7 and 8, we carry out further simulation studies as in Table 2.1 for the case $\rho < 1$ but $[m\rho] = 50$, unequal sample sizes and different choices of θ_1 values, respectively, while Appendix.9 examines assumption (1), the independence of SNPs, in Section 2.2.1.

Table 2.2: Growth rate of the $\widehat{PCC(n)}$ as $Size = 2n$ (n for C_1, n for C_2). $h = \theta_1 - \theta_2$, and m (the number of independent SNPs) vary; $\theta_1 = 0.3, \rho = 1$ and $\alpha = 0.01$.

		$\widehat{PCC(n)}$				
h	m	$2n = 60$	$2n = 200$	$2n = 1000$	$2n = 2000$	$\widehat{PCC(\infty)}$
0.01	10	0.5035	0.5035	0.5039	0.5042	0.5197
0.01	50	0.5049	0.505	0.5059	0.5068	0.5437
0.01	100	0.506	0.5062	0.5075	0.5089	0.5616
0.01	200	0.5077	0.508	0.5099	0.512	0.5868
0.05	10	0.5216	0.5281	0.5606	0.5866	0.5997
0.05	50	0.5336	0.5506	0.6289	0.6858	0.7128
0.05	100	0.5435	0.5684	0.6783	0.7529	0.7864
0.05	200	0.558	0.5941	0.7432	0.833	0.8691
0.1	10	0.5656	0.6164	0.6997	0.7004	0.7004
0.1	50	0.6203	0.7356	0.8776	0.8784	0.8784
0.1	100	0.6621	0.8118	0.9498	0.9504	0.9504
0.1	200	0.7192	0.8937	0.9899	0.9901	0.9901
0.2	10	0.8048	0.8836	0.8845	0.8845	0.8845
0.2	50	0.9681	0.9958	0.9959	0.9959	0.9959
0.2	100	0.9954	0.9999	0.9999	0.9999	0.9999
0.2	200	0.9999	1	1	1	1

Next, we turn our attention to determining the smallest n such that $f(n) = \widehat{PCC(\infty)} - \widehat{PCC(n)} - \gamma < 0$, for any specific value of γ. We use the following algorithm to determine such an n: (i) Let $n = n_S$ and n_L such that $f(n_S) > 0$ and $f(n_L) < 0$, and set $n_M = [(n_S + n_L)/2]$. To begin the algorithm select a small n_S and a large n_L; (ii) If $f(n_M)f(n_S) < 0$, then reset $n_L = n_M$; else, reset $n_S = n_M$. In either case, return to step (i), unless $n_L - n_S \leq 1$, in which case, the smallest sample $n = n_L$; (iii) Use the smallest (total) sample of size $2n_L$, with half from each class, C_1 and C_2. We implemented this algorithm for each value of h, m and significance level α for the Wald test. Table 2.3 shows the determined sample sizes for each combination of parameters. From Table 2.3 and the plot of these values in Figure 2.1, it is evident that the required sample size reduces as h increases, as expected. Also, since $\rho = 1$,

all the SNPs under consideration are significant. Therefore, as α increases, many significant SNPs are included in the classifier, which improves the predictive ability of the classifier. Hence, $f(n) < 0$ for smaller sample sizes, as shown in Table 2.3. However, the effect of m on the determined sample sizes is less clear. When h is large, say $h = 0.2$, then the required sample size reduces as m becomes large. Whereas, when h is small, say $h = 0.01$ or 0.05, the reverse is true as m becomes large.

Table 2.3: Sample size determination using the algorithm in Section 2.3.1. The $Size = 2n$ (n for C_1, n for C_2) required to satisfy: $\widehat{PCC(\infty)} - \widehat{PCC(n)} < \gamma$ ($= 0.01$). Here, $\theta_1 = 0.3, h = \theta_1 - \theta_2$, $m =$ the number of independent SNPs, $\rho = 1$ and $\alpha = 0.01$.

		Sample Size			
α	h	$m = 10$	$m = 50$	$m = 100$	$m = 200$
0.01	0.01	18,900	41,976	49,630	56,536
0.01	0.05	2,208	2,710	2,842	2,872
0.01	0.1	612	660	616	490
0.01	0.2	132	84	50	30
0.05	0.01	9,548	27,278	33,512	39,226
0.05	0.05	1,538	1,962	2,074	2,100
0.05	0.1	440	482	444	338
0.05	0.2	96	56	30	14
0.1	0.01	5,936	20,976	26,484	31,590
0.1	0.05	1,242	1,626	1,728	1,752
0.1	0.1	364	402	368	272
0.1	0.2	80	44	22	10

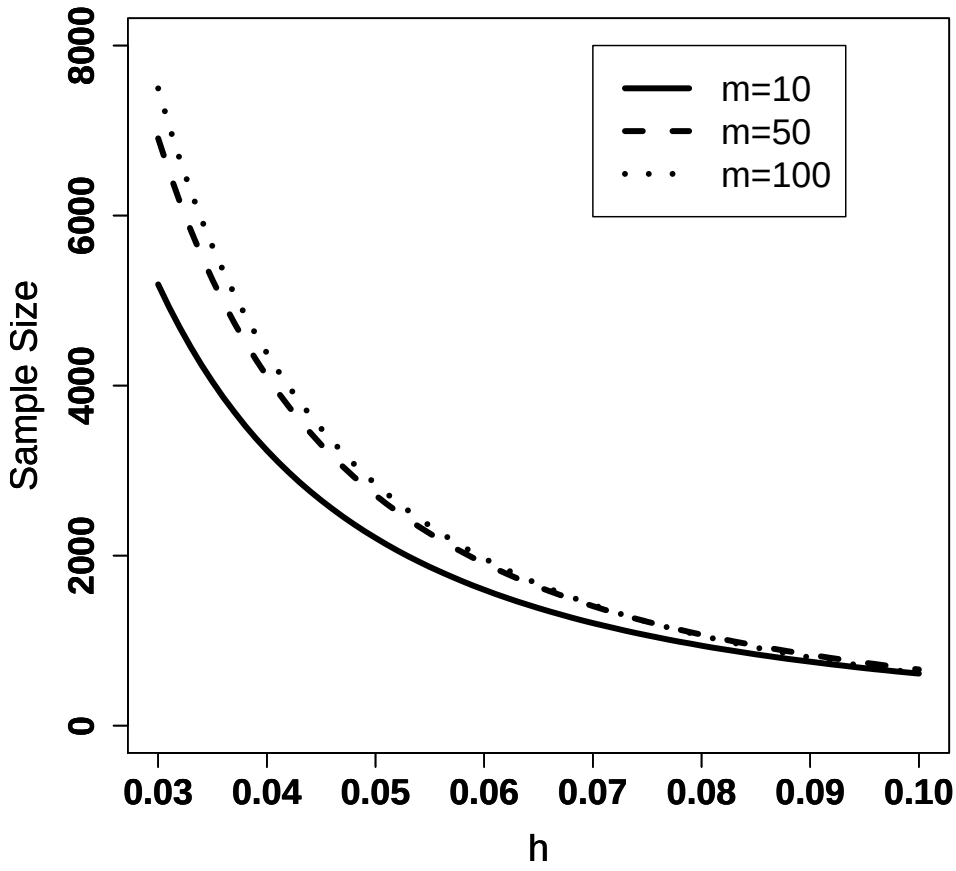

Figure 2.1: *Size* vs h. Here, $\theta_1 = 0.3, \gamma = 0.01$, $\rho = 1$, and $\alpha = 0.01$.

2.3.2 *Application to the HapMap Data*

The aim of the International HapMap Project is to develop a haplotype map of the human genome, the so-called HapMap, which will describe the common patterns of human DNA sequence variation. Discovering the DNA sequence variants that contribute to common disease risk offers one of the best opportunities for understanding the complex causes of disease in humans (`http://hapmap.ncbi.nlm.nih.gov/abouthapmap.html`). The HapMap data consists of four populations with about $p = 1.2 \times 10^6$ SNPs. Here, we consider the following two classes: C_1 – the Han Chinese individuals from Beijing (CHB) with 137 subjects and C_2 – the Japanese individuals from Tokyo (JTP) with 113 subjects, for a total of 250 subjects. Suppose we set a value for $\gamma = \gamma_0$, say, we will now illustrate how to construct a linear classifier of the form given in Section 2.2.3 based on pair-wise independent SNPs, and determine a (total) sample size such that $\widehat{PCC(\infty)} - \widehat{PCC(n)} < \gamma_0$.

Based on all the 250 available subjects, we extracted pair-wise independent SNPs using the following procedure. Suppose L is a set of SNPs, then (I) form the set S with one SNP from L and update S after next step; (II) from the remaining SNPs in L, choose one SNP that is independent of every SNP in S using Kendall's τ coefficient as a test statistic to test pair-wise independence, and then add this new SNP to S. Here, we concluded independence if the Kendall's τ-value < 0.1; (III) Repeat (II) until each remaining SNP in L is correlated with at least one SNP in S. This procedure yielded a set S with $m = 51$ pair-wise independent SNPs, and with these we built our linear classifier.

Next, we set $\rho = 1$ so that $m = l = 51$. Recall that $\vec{\theta_1} = (\theta_{1,1}, \ldots, \theta_{1,l})'$ and $\vec{\theta_2} = (\theta_{2,1}, \ldots, \theta_{2,l})'$. We estimated $\vec{\theta_1}$ and $\vec{\theta_2}$ using the maximum likelihood (ML) estimates (based on 137 and 113 subjects, respectively) given in Appendix.3. We substituted these ML estimates, $\hat{\pi}_1 = 0.548$, and $\hat{h}_j$ into the expressions for $\widehat{PCC(\infty)}$ and $\widehat{PCC(n)}$ given in (2.2) and (2.3), respectively, and used our algorithm (see Section 2.3.1) to determine the total sample size satisfying the criterion, $\widehat{PCC(\infty)} - \widehat{PCC(n)} < \gamma$, for various values of γ. Figure 2.2 shows a plot of required total sample size versus γ, when $\alpha = 0.1$ and 0.2. For instance, if

$\gamma = 0.05$ and $\alpha = 0.1$, then the total sample of size ≈ 166 (83 for CBH and 83 for JPT) satisfies the criterion. The fact that we have an approximate expression for $PCC(n)$ corresponding to our classifier enables us to algorithmically determine a total sample which satisfies the criterion, for any given γ. However, although SVM is a competitive classifier, we cannot determine a total sample size that satisfies the criterion for a given γ because SVM does not have an approximate (or an exact) expression for its $PCC(n)$. This highlights the usefulness of our linear classifier and the approximate formula for its $PCC(n)$.

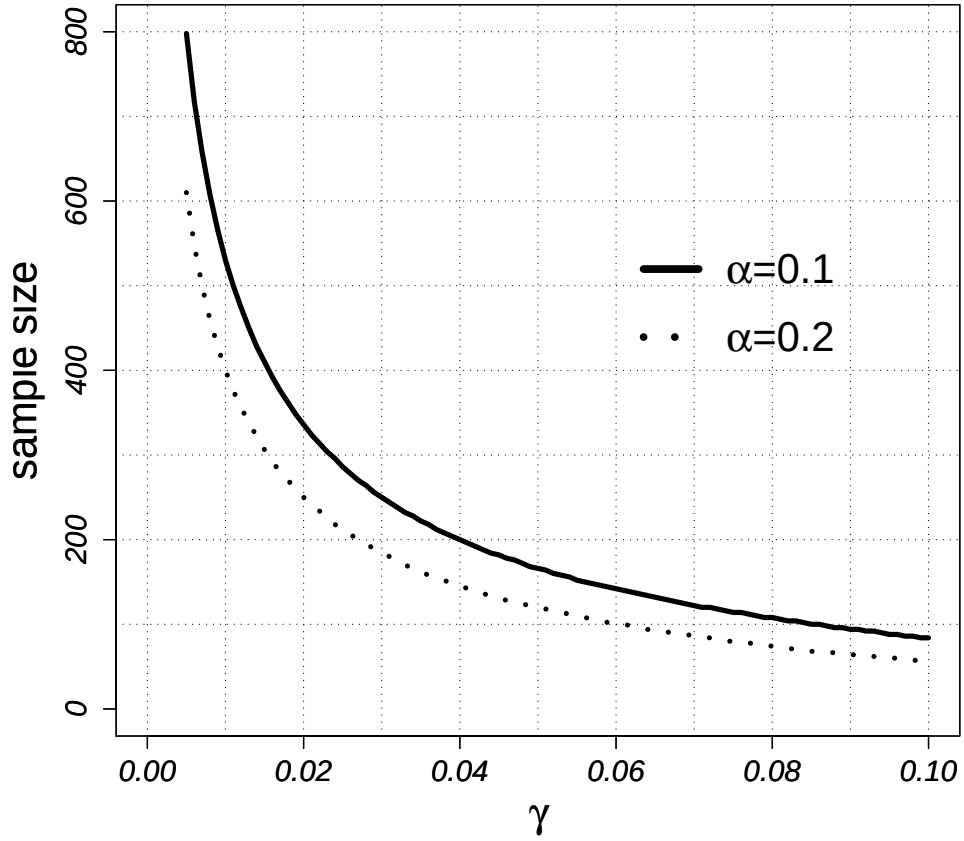

Figure 2.2: Sample Size versus γ for the HapMap data.

2.4 Discussion

For a SNP-based classifier for two populations, a criterion for sample size determination based on probability of correct classification is proposed and applied to the HapMap data on Chinese and Japanese population. The postulated parametric model for the SNP data allows us to substitute the maximum likelihood estimates of parameters into approximate formulas for probability of correct classification and use a simple computational algorithm to obtain the required sample size. In fact, a major advantage of our method over the learning curve methods such as the SVM or NN is that, given a threshold γ and a sample, we can algorithmically obtain a total sample size required to satisfy the criterion. Whereas, the SVM and NN would only be able to compute the probability of correct classification, but not provide a total sample size required to satisfy a given criterion. Our method is transparent and easy to use, as illustrated in our HapMap data analysis.

If the data at hand is small, then the parameter estimates may be biased. In fact, in some instances, there may be no sample at hand to compute parameter estimates. In such cases, it may be better to adopt a sequential sampling method (Fu *and others*, 2005) for sample size determination. Also, in population classification problems, there may be more than two classes, e.g., HapMap data. Generalizations of our method to multiple populations will be considered elsewhere.

2.5 Appendices

In the following sections, the equation numbers without the prefix, A, correspond to those in the main article.

Appendix.1 Derivation of the optimal classifier

The classification rule in Section 2.2.2 is equivalent to: "Classify $\vec{x}$ to C_1" if $\log(\frac{\tau_1(\vec{\theta_1}|\vec{x})}{\tau_2(\vec{\theta_2}|\vec{x})}) > 0$, which is same as

$$\sum_{j=1}^{l} x_j \log\left(\frac{\theta_{1,j}(1-\theta_{2,j})}{\theta_{2,j}(1-\theta_{1,j})}\right) > \log\left(\frac{(1-\pi_1)}{\pi_1}\right) + 2\sum_{j=1}^{l}\log\left(\frac{1-\theta_{2,j}}{1-\theta_{1,j}}\right).$$

This reduces to $\sum_{j=1}^{l} b_j x_j > K$ for b_j and K defined in the *optimal* classifier.

Appendix.2 Approximation for $PCC(\infty)$

The PCC for the optimal classifier in Section 2.2.2 is:

$$PCC(\infty) \;=\; \pi_1 P(\sum_{j=1}^{l} b_j x_j > K | \vec{x} \in C_1) + (1-\pi_1) P(\sum_{j=1}^{l} b_j x_j < K | \vec{x} \in C_2). \quad \text{(A.1)}$$

To obtain an approximation to each of these terms, let $Y_j = b_j x_j$ and note that $|Y_j| \leq M$, for some $M > 0$, because $|x_j| \leq 2$ and $|b_j| \leq log(99)$ since $\theta_1, \theta_2 \in (0.01, 0.5)$. Therefore, $\{Y_j\}$ is a sequence of independent and bounded random variables with $E(Y_j) = 2b_j\theta_{k,j}$ and $Var(Y_j) = 2b_j^2\theta_{k,j}(1-\theta_{k,j})$, when $\vec{x} \in C_k$ for $k = 1, 2$. Furthermore, since $\theta_{k,j}(1-\theta_{k,j}) > 0.01 \times 0.5$ for $k = 1, 2$, by assumption (3) in Section 2.2.1 we have

$$\sum_{j=1}^{l} Var(Y_j) = 2\sum_{j=1}^{l} b_j^2\theta_{k,j}(1-\theta_{k,j}) > (0.01)\sum_{j=1}^{l} b_j^2 \to \infty,$$

as $l \to \infty$. Therefore, by Theorem 27-4 (also see Problem 27.4) of Billingsley (1995), we have for each $k = 1, 2$ that

$$\frac{\sum_{j=1}^{l} b_j x_j - 2\sum_{j=1}^{l} b_j\theta_{k,j}}{\sqrt{2\sum_{j=1}^{l}[b_j^2\theta_{k,j}(1-\theta_{k,j})]}} \xrightarrow{d} N(0,1) \;\text{ as }\; l \to \infty. \quad \text{(A.2)}$$

Therefore, from (A.1) and (A.2), we have that

$$PCC(\infty) \approx \pi_1\Phi\left(\frac{(2\sum_{j=1}^{l} b_j\theta_{1,j} - K)}{\sqrt{2\sum_{j=1}^{l}[b_j^2\theta_{1,j}(1-\theta_{1,j})]}}\right) + (1-\pi_1)\Phi\left(\frac{(K - 2\sum_{j=1}^{l} b_j\theta_{2,j})}{\sqrt{2\sum_{j=1}^{l}[b_j^2\theta_{2,j}(1-\theta_{2,j})]}}\right).$$

Appendix.3 Wald test and its power function

For each X_j satisfying assumption (2) in Section 2.2.1, let $n_{1k} = \sum_{j=1}^{n_k} I_{\{x_j=0\}}$, $n_{2k} = \sum_{j=1}^{n_k} I_{\{x_j=1\}}$ and $n_{3k} = \sum_{j=1}^{n_k} I_{\{x_j=2\}}$ with $\sum_{i=1}^{3} n_{ik} = n_k$ for $k = 1, 2$. Since we are interested in developing Wald test for testing $H_0 : \theta_{1,j} = \theta_{2,j}$ versus $H_1 : \theta_{1,j} \neq \theta_{2,j}$ for each j, for notational convenience, let $\theta_{1,j} = \theta_1$ and $\theta_{2,j} = \theta_2$ for the derivations below. Then, by (2.1) of the article and independence of the two classes, the likelihood function for a total sample of size $2n(= n_1 + n_2)$ is :

$$L(\theta_1, \theta_2) = \prod_{k=1}^{2} [(1-\theta_k)^2]^{n_{1k}} [2\theta_k(1-\theta_k)]^{n_{2k}} [\theta_k^2]^{n_{3k}}.$$

Maximizing the log-likelihood, $logL(\theta_1, \theta_2)$, with respect to (θ_1, θ_2), it can be shown that the maximum likelihood estimator (MLE) of θ_1 and θ_2 are, respectively:

$$\hat{\theta_1} = \frac{n_{21} + 2n_{31}}{2n} \quad \text{and} \quad \hat{\theta_2} = \frac{n_{22} + 2n_{32}}{2n}. \tag{A.3}$$

Also, the Fisher information matrix at (θ_1, θ_2) for $n_1 = n_2 = 1$ is $I(\theta_1, \theta_2) = \begin{pmatrix} \frac{2}{\theta_1(1-\theta_1)} & 0 \\ 0 & \frac{2}{\theta_2(1-\theta_2)} \end{pmatrix}$. By the asymptotic normality of the MLE, it follows that $\sqrt{n}(\hat{\theta_1} - \theta_1, \hat{\theta_2} - \theta_2)' \xrightarrow{d} N_2(\mathbf{0}, I^{-1}(\theta_1, \theta_2))$. Now, since $g(\theta_1, \theta_2) = \theta_1 - \theta_2$ is differentiable at (θ_1, θ_2), it follows from the delta method that $\sqrt{n}[g(\hat{\theta_1}, \hat{\theta_2}) - g(\theta_1, \theta_2)] \xrightarrow{d} N(0, \frac{\theta_1(1-\theta_1)+\theta_2(1-\theta_2)}{2})$. Therefore, under $H_0 : \theta_1 = \theta_2$, the Wald test statistic

$$Q_2 = \frac{2n(\hat{\theta_1} - \hat{\theta_2})^2}{\hat{\theta_1}(1-\hat{\theta_1}) + \hat{\theta_2}(1-\hat{\theta_2})} \xrightarrow{d} \chi_1^2 \quad \text{as} \quad n \to \infty,$$

where χ_1^2 has chi-square distribution with 1 degree of freedom. However, under $H_a : \theta_1 \neq \theta_2$, say $\theta_1 - \theta_2 = h$, it follows from the above arguments that $Q_2 \xrightarrow{d} \chi_1^2(\lambda^2)$, where $\chi_1^2(\lambda^2)$ has non-central chi-square distribution with the non-centrality parameter, $\lambda^2 = 2nh^2/[\theta_1(1-\theta_1) + (\theta_1 - h)(1 - \theta_1 + h)]$. Therefore, the power of the Wald test (when $\theta_1 - \theta_2 = h \neq 0$) is:

$$1 - \beta(n, h) \approx P\left(\chi_1^2(\lambda^2) > \chi_{1,(1-\alpha)}^2\right),$$

where $\chi_{1,(1-\alpha)}^2$ is the $(1-\alpha)$ percentile of χ_1^2. For ease of presentation, we had suppressed the subscript j. For each $j = 1, \ldots, m$, the power of the Wald test for $H_0 : \theta_{1,j} = \theta_{2,j}$ versus $H_1 : \theta_{1,j} \neq \theta_{2,j}$ at $\theta_{1,j} = \theta_{2,j} + h_j$ is denoted by $1 - \beta_j(n, h_j)$.

Appendix.4 Approximation for $PCC(n)$

The PCC for the linear classifier, $\sum_{j=1}^{m} w_{j,n}x_j$, is

$$PCC(n) \;=\; \pi_1 P(\sum_{j=1}^{m} \hat{b}_j w_{j,n}x_j > K|\vec{x} \in C_1) + (1-\pi_1)P(\sum_{j=1}^{m} \hat{b}_j w_{j,n}x_j < K|\vec{x} \in C_2). \tag{A.4}$$

Note from the definition of $w_{j,n}$'s in Section 2.2.3 and assumption (3) in Section 2.2.1 that

$$P(rejectH_0) = \rho[1-\beta_j(n,h_j)] + (1-\rho)\alpha = \tilde{\eta}_j(\alpha,\rho,n,h_j)$$

$$E(w_{j,n}) = \tilde{\eta}_j(\alpha,\rho,n,h_j)$$

$$E(w_{j,n}^2) = \tilde{\eta}_j(\alpha,\rho,n,h_j),$$

where α and $[1-\beta_j(n,h_j)]$ are the level and power, respectively, of the Wald test and $\tilde{\eta}_j(\alpha,\rho,n,h_j) = \tilde{\eta}_j$ is defined in Section 2.2.3; also, see Appendix.3.

Now, let $\tilde{Y}_{j,n} = \hat{b}_j w_{j,n}x_j$ and note that $|\tilde{Y}_{j,n}| \le 2\log(99)$. For each n, $\{\tilde{Y}_{j,n}\}$ is a sequence of independent and bounded random variables with $E(\tilde{Y}_{j,n}) \approx 2\theta_{k,j}\tilde{\eta}_j b_j$ and by the fact that $V(Y_{j,n}) = [V(Y_{j,n}|w_{j,n})] + V[E(Y_{j,n}|w_{j,n})]$ we have

$$Var(\tilde{Y}_{j,n}) \approx b_j^2[2\theta_{k,j}(1-\theta_{k,j})\tilde{\eta}_j + 4\theta_{k,j}^2\tilde{\eta}_j(1-\tilde{\eta}_j)],$$

when $\vec{x} \in C_k$ for $k = 1,2$. Furthermore, since $\theta_{k,j}(1-\theta_{k,j}) > 0.005$ for $k = 1,2$ and $\tilde{\eta}_j(\alpha,\rho,n,h_j) > (1-\rho)\alpha$ (see above), we have

$$\sum_{j=1}^{m} Var(\tilde{Y}_{j,n}) > (0.01)[(1-\rho)\alpha]m \;\to\; \infty,$$

as $m \to \infty$. Therefore, once again, by Theorem 27.2 of Billingsley (1979), we have for each $k = 1,2$ that

$$\frac{\sum_{j=1}^{m} w_{j,n}\hat{b}_j x_j - 2\sum_{j=1}^{m}\theta_{k,j}\tilde{\eta}_j b_j}{\sqrt{\sum_{j=1}^{m} b_j^2[2\theta_{k,j}(1-\theta_{k,j})\tilde{\eta}_j + 4\theta_{k,j}^2\tilde{\eta}_j(1-\tilde{\eta}_j)]}} \xrightarrow{d} N(0,1), \tag{A.5}$$

as $m \to \infty$. The normal approximations in (2.3) follows from (A.4) and (A.5).

Appendix.5 Maximum value of K

Note that the approximations for $PCC(\infty)$ and $PCC(n)$ in (2.2) and (2.3), respectively, are of the form

$$g(x) = \pi_1 \Phi(\frac{a-x}{b}) + (1-\pi_1)\Phi(\frac{x-c}{d})$$

for some constants a, b, c and d with $b \neq d$. Using calculus, it can be shown that g attains its maximum when $x = x_0$, where

$$x_0 = \frac{(ad^2 - cb^2) - bd\sqrt{(a-c)^2 - 2(d^2-b^2)log(\frac{(1-\pi_1)b}{\pi_1 d})}}{d^2 - b^2}.$$

Appendix.6 The case $\rho < 1$

In the Tables 2.1 to 2.3 of the article, we set the percentage ρ of SNPs with marginal effect on case-control equal to 1, that is, $\rho = 1$. Suppose we vary the values of ρ, but set $[m\rho] = 50$ in assumption (3) of Section 2.2.1, so that $m \approx 50/\rho$. This means, when ρ is small, say 0.1, there is a high proportion of insignificant SNPs compared to significant ones for each subject. Nevertheless, as in the case of $\rho = 1$ in Table 2.1 of the article, the $\widehat{Bias} = \widehat{PCC(n)}\text{MC} - \widehat{PCC(n)}$ values in Table 2.4 (see below) continue to be negligible in most cases. This, once again, validates the accuracy of approximations in (2.3) for changing values of ρ. However, when ρ is small and the total sample size is moderate, the difference, $\widehat{PCC(\infty)} - \widehat{PCC(n)}$, is large. This is because, when ρ is small, the number of insignificant SNPs is large, which increases the probability of type I and type II errors, which in turn impacts the value of $\widehat{PCC(n)}$. As before, Table 2.4 also gives the $PCC(n)$ values for SVM, which are once again close to $\widehat{PCC(n)}MC$ values of our linear classifier.

Table 2.4: Performance of Optimal and Linear classifiers when $\rho < 1$. The values of $\widehat{PCC(n)}$ and $\widehat{PCC(n)}$MC are close to each other for various model specifications. Here, $\theta_1 = 0.3, h = \theta_1 - \theta_2$, m is the number of SNP's, $Size = 2n$ (n for C_1, n for C_2), and $\alpha = 0.01$. The percentage ρ of significant SNPs varies, but $\lfloor m\rho \rfloor = 50$.

h	ρ	$Size$	$\widehat{PCC(\infty)}$	$\widehat{PCC(n)}$	$\widehat{PCC(n)}$MC	$\widehat{Bias}$	SVM_MC
0.05	0.1	100	0.7128	0.5387	0.5183	-0.0204	0.5175
0.05	0.1	200	0.7128	0.5506	0.5331	-0.0175	0.5302
0.05	0.1	2000	0.7128	0.6858	0.6767	-0.0091	0.664
0.05	0.2	100	0.7128	0.5387	0.5271	-0.0116	0.5253
0.05	0.2	200	0.7128	0.5506	0.5502	-0.0004	0.5436
0.05	0.2	2000	0.7128	0.6858	0.705	0.0192	0.6948
0.05	0.5	100	0.7128	0.5387	0.5296	-0.0091	0.5285
0.05	0.5	200	0.7128	0.5506	0.5495	-0.0011	0.55
0.05	0.5	2000	0.7128	0.6858	0.6952	0.0094	0.6835
0.1	0.1	100	0.8784	0.6577	0.6445	-0.0132	0.6414
0.1	0.1	200	0.8784	0.7356	0.7728	0.0372	0.7654
0.1	0.1	2000	0.8784	0.8784	0.8864	0.008	0.879
0.1	0.2	100	0.8784	0.6577	0.6468	-0.0109	0.6353
0.1	0.2	200	0.8784	0.7356	0.738	0.0024	0.7281
0.1	0.2	2000	0.8784	0.8784	0.8774	-0.001	0.8702
0.1	0.5	100	0.8784	0.6577	0.6816	0.0239	0.6692
0.1	0.5	200	0.8784	0.7356	0.7517	0.0161	0.7428
0.1	0.5	2000	0.8784	0.8784	0.8962	0.0178	0.89
0.2	0.1	100	0.9959	0.9908	0.974	-0.0168	0.9741
0.2	0.1	200	0.9959	0.9958	0.9912	-0.0046	0.991
0.2	0.1	2000	0.9959	0.9959	0.9948	-0.0011	0.9937
0.2	0.2	100	0.9959	0.9908	0.9782	-0.0126	0.9784
0.2	0.2	200	0.9959	0.9958	0.9934	-0.0024	0.9935
0.2	0.2	2000	0.9959	0.9959	0.991	-0.0049	0.9889
0.2	0.5	100	0.9959	0.9908	0.989	-0.0018	0.9892
0.2	0.5	200	0.9959	0.9958	0.9809	-0.0149	0.9774
0.2	0.5	2000	0.9959	0.9959	0.9883	-0.0076	0.9862

Appendix.7 Unequal sample sizes

In the Tables 2.1 to 2.3 of the article, we considered the two class sample sizes to be the same, that is, $n_1 = n_2$. Here, we consider unequal sample sizes such that n_1 and n_2 are

different fractions of the total sample size. For example, in Table 2.5, we let $n_1 : n_2 = 2 : 1$, and in Table 2.6, we let $n_1 : n_2 = 5 : 1$. In both case, the Bias is small, as in Table 2.1 of the article. This shows that our sample size determination method is also valid for unequal (class) sample sizes.

Table 2.5: Performance of Optimal and Linear classifiers when $n_1/n_2 = 2$. The values of $\widehat{PCC(n)}$ and $\widehat{PCC(n)}$MC are close to each other for various model specifications. Here, $\theta_1 = 0.3, h = \theta_1 - \theta_2$, $Size = n_1 + n_2$ ($n_1 : n_2 = 2 : 1$), m is the number of independent SNPs, $\alpha = 0.01$ is the significant level for Wald tests, and $\rho = 1$ is the percentage of the significant SNPs.

h	m	$Size$	$\widehat{PCC(\infty)}$	$\widehat{PCC(n)}$	$\widehat{PCC(n)}$MC	$\widehat{Bias}$
0.01	10	60	0.6667	0.6667	0.6473	-0.0194
0.01	10	200	0.6667	0.6667	0.6637	-0.003
0.01	10	2000	0.6667	0.6667	0.6665	-0.0002
0.01	50	60	0.6668	0.6667	0.6375	-0.0292
0.01	50	200	0.6668	0.6667	0.6582	-0.0085
0.01	50	2000	0.6668	0.6667	0.6653	-0.0014
0.01	200	60	0.6724	0.6667	0.6065	-0.0602
0.01	200	200	0.6724	0.6667	0.6216	-0.0451
0.01	200	2000	0.6724	0.6667	0.6565	-0.0102
0.05	10	60	0.6811	0.6674	0.657	-0.0104
0.05	10	200	0.6811	0.6679	0.6583	-0.0096
0.05	10	2000	0.6811	0.6777	0.6714	-0.0063
0.05	50	60	0.751	0.6684	0.625	-0.0434
0.05	50	200	0.751	0.6708	0.6473	-0.0235
0.05	50	2000	0.751	0.7333	0.7289	-0.0044
0.05	200	60	0.8815	0.6724	0.6403	-0.0321
0.05	200	200	0.8815	0.6835	0.6732	-0.0103
0.05	200	2000	0.8815	0.8523	0.853	0.0007
0.2	10	60	0.9034	0.8515	0.8183	-0.0332
0.2	10	200	0.9034	0.9032	0.8853	-0.0179
0.2	10	2000	0.9034	0.9034	0.8888	-0.0146
0.2	50	60	0.9965	0.9783	0.9643	-0.014
0.2	50	200	0.9965	0.9965	0.9934	-0.0031
0.2	50	2000	0.9965	0.9965	0.9953	-0.0012
0.2	200	60	1	1	0.9817	-0.0183
0.2	200	200	1	1	0.9994	-0.0006
0.2	200	2000	1	1	0.9999	-0.0001

Table 2.6: Performance of Optimal and Linear classifiers when $n_1/n_2 = 5$. The values of $\widehat{PCC(n)}$ and $\widehat{PCC(n)}$MC are close to each other for various model specifications. Here, $\theta_1 = 0.3, h = \theta_1 - \theta_2$, $Size = n_1 + n_2$ ($n_1 : n_2 = 5 : 1$), m is the number of independent SNPs, $\alpha = 0.01$ is the significant level for Wald tests, and $\rho = 1$ is the percentage of the significant SNPs.

h	m	$Size$	$\widehat{PCC(\infty)}$	$\widehat{PCC(n)}$	$\widehat{PCC(n)}$MC	$\widehat{Bias}$
0.01	10	60	0.8333	0.8333	0.7537	-0.0796
0.01	10	200	0.8333	0.8333	0.8004	-0.0329
0.01	10	2000	0.8333	0.8333	0.8321	-0.0012
0.01	50	60	0.8333	0.8333	0.803	-0.0303
0.01	50	200	0.8333	0.8333	0.816	-0.0173
0.01	50	2000	0.8333	0.8333	0.8324	-0.0009
0.01	200	60	0.8333	0.8333	0.7768	-0.0565
0.01	200	200	0.8333	0.8333	0.782	-0.0513
0.01	200	2000	0.8333	0.8333	0.8294	-0.0039
0.05	10	60	0.8338	0.8333	0.8237	-0.0096
0.05	10	200	0.8338	0.8334	0.823	-0.0104
0.05	10	2000	0.8338	0.8337	0.8324	-0.0013
0.05	50	60	0.8497	0.8334	0.7958	-0.0376
0.05	50	200	0.8497	0.8334	0.8068	-0.0266
0.05	50	2000	0.8497	0.8446	0.8388	-0.0058
0.05	200	60	0.9179	0.8335	0.811	-0.0225
0.05	200	200	0.9179	0.8344	0.8229	-0.0115
0.05	200	2000	0.9179	0.9017	0.8949	-0.0068
0.2	10	60	0.9381	0.9207	0.8848	-0.0359
0.2	10	200	0.9381	0.9381	0.9159	-0.0222
0.2	10	2000	0.9381	0.9381	0.9232	-0.0149
0.2	50	60	0.9976	0.992	0.969	-0.023
0.2	50	200	0.9976	0.9976	0.9933	-0.0043
0.2	50	2000	0.9976	0.9976	0.9962	-0.0014
0.2	200	60	1	1	0.9762	-0.0238
0.2	200	200	1	1	0.9984	-0.0016
0.2	200	2000	1	1	0.9999	-0.0001

Appendix.8 Different minor allele frequencies of class 1 (θ_1)

In Table 2.1 of the article, we set $\theta_1 = 0.3$ and carried out our simulation study by varying h. To show that our method is not sensitive to the choice of θ_1 values, we set $\theta_1 = 0.25$ and 0.5 and for each of these values, we once again carried out the simulation study in Section 2.3.1 by varying h. These results are given in Tables 2.7 and 2.8, respectively. Once again, the Bias values are small, as in Table 2.1 of the article.

Table 2.7: Performance of Optimal and Linear classifiers when $\theta_1 = 0.25$. The values of $\widehat{PCC(n)}$ and $\widehat{PCC(n)}$MC are close to each other for various model specifications. Here, $\theta_1 = 0.25, h = \theta_1 - \theta_2$, $Size = n_1 + n_2$ (1/2 for C_1, 1/2 for C_2), m is the number of independent SNPs, $\alpha = 0.01$ is the significant level for Wald tests, and $\rho = 1$ is the percentage of the significant SNPs.

h	m	$Size$	$\widehat{PCC(\infty)}$	$\widehat{PCC(n)}$	$\widehat{PCC(n)}$MC	$\widehat{Bias}$
0.01	10	60	0.5209	0.504	0.5022	-0.0018
0.01	10	200	0.5209	0.5041	0.4995	-0.0046
0.01	10	2000	0.5209	0.505	0.5032	-0.0018
0.01	50	60	0.5464	0.5056	0.4947	-0.0109
0.01	50	200	0.5464	0.5058	0.5034	-0.0024
0.01	50	2000	0.5464	0.508	0.5084	0.0004
0.01	200	60	0.592	0.5087	0.5093	0.0006
0.01	200	200	0.592	0.5092	0.5058	-0.0034
0.01	200	2000	0.592	0.5141	0.515	0.0009
0.05	10	60	0.6069	0.5256	0.5043	-0.0213
0.05	10	200	0.6069	0.5338	0.5263	-0.0075
0.05	10	2000	0.6069	0.5982	0.5964	-0.0018
0.05	50	60	0.7263	0.5398	0.5172	-0.0226
0.05	50	200	0.7263	0.561	0.5619	0.0009
0.05	50	2000	0.7263	0.7085	0.7098	0.0013
0.05	200	60	0.8851	0.5685	0.559	-0.0095
0.05	200	200	0.8851	0.6134	0.6252	0.0118
0.05	200	2000	0.8851	0.8633	0.8651	0.0018
0.2	10	60	0.9211	0.8678	0.8392	-0.0286
0.2	10	200	0.9211	0.9208	0.9109	-0.0099
0.2	10	2000	0.9211	0.9211	0.9169	-0.0042
0.2	50	60	0.999	0.9916	0.981	-0.0106
0.2	50	200	0.999	0.999	0.9965	-0.0025
0.2	50	2000	0.999	0.999	0.9983	-0.0007
0.2	200	60	1	1	0.9853	-0.0147
0.2	200	200	1	1	0.9994	-0.0006
0.2	200	2000	1	1	1	0

Table 2.8: Performance of Optimal and Linear classifiers when $\theta_1 = 0.5$. The values of $\widehat{PCC(n)}$ and $\widehat{PCC(n)}$MC are close to each other for various model specifications. Here, $\theta_1 = 0.5, h = \theta_1 - \theta_2$, $Size = n_1 + n_2$ (1/2 for C_1, 1/2 for C_2), m is the number of independent SNPs, $\alpha = 0.01$ is the significant level for Wald tests, and $\rho = 1$ is the percentage of the significant SNPs.

h	m	$Size$	$\widehat{PCC(\infty)}$	$\widehat{PCC(n)}$	$\widehat{PCC(n)}$MC	$\widehat{Bias}$
0.01	10	60	0.5178	0.5023	0.4992	-0.0031
0.01	10	200	0.5178	0.5024	0.5005	-0.0019
0.01	10	2000	0.5178	0.5028	0.5017	-0.0011
0.01	50	60	0.5398	0.5033	0.495	-0.0083
0.01	50	200	0.5398	0.5034	0.5004	-0.003
0.01	50	2000	0.5398	0.5045	0.5035	-0.001
0.01	200	60	0.5793	0.5054	0.4972	-0.0082
0.01	200	200	0.5793	0.5056	0.5046	-0.001
0.01	200	2000	0.5793	0.5081	0.5122	0.0041
0.05	10	60	0.5887	0.5138	0.5048	-0.009
0.05	10	200	0.5887	0.5177	0.5161	-0.0016
0.05	10	2000	0.5887	0.5622	0.5737	0.0115
0.05	50	60	0.6919	0.5219	0.5193	-0.0026
0.05	50	200	0.6919	0.532	0.5513	0.0193
0.05	50	2000	0.6919	0.636	0.6605	0.0245
0.05	200	60	0.842	0.5381	0.5483	0.0102
0.05	200	200	0.842	0.5598	0.5902	0.0304
0.05	200	2000	0.842	0.7564	0.7969	0.0405
0.2	10	60	0.825	0.6676	0.7005	0.0329
0.2	10	200	0.825	0.8044	0.8034	-0.001
0.2	10	2000	0.825	0.825	0.8197	-0.0053
0.2	50	60	0.9816	0.824	0.875	0.051
0.2	50	200	0.9816	0.9721	0.9715	-0.0006
0.2	50	2000	0.9816	0.9816	0.9792	-0.0024
0.2	200	60	1	0.9676	0.9587	-0.0089
0.2	200	200	1	0.9999	0.9977	-0.0022
0.2	200	2000	1	1	0.9997	-0.0003

Appendix.9 $\widehat{PCC(n)}$MC calculations for dependent SNPs

In Section 2.2.1 of the article, we assumed that the data vector consists of m independent SNPs. To determine the effect of relaxing the assumption of "independence" between m SNPs, we calculated the Monte Carlo based estimates, $\widehat{PCC(n)}$MC, by randomly generating SNP datasets with different correlations between them. More specifically, we randomly generated values (0,1, or 2) for the first SNP with $n = 60$ and 200. To generate values for the second SNP such that it has, say, 10% correlation with the first SNP, we generated a uniform random number U for each observation. If $U < 0.1$, then we assigned the same value as for first SNP for that observation; otherwise, we randomly generated a value (0,1 or 2) using the θ value. Similarly, we constructed the values for m SNPs. In Table 2.9 below, we give $\widehat{PCC(n)}$MC values for various specifications of $h, m,$ and n, and correlations (between SNPs) ranging from 0% to 50%. This table shows that the $\widehat{PCC(n)}$MC values are similar for almost all correlations, indicating that correlation between SNPs does not severely impact the values of $\widehat{PCC(n)}$MC.

Table 2.9: Performance of $\widehat{PCC(n)}$MC when SNPs are correlated. Here, $\theta_1 = 0.3, h = \theta_1 - \theta_2$, $Size = n_1 + n_2$ (1/2 for C_1, 1/2 for C_2), m is the number of SNPs, $\alpha = 0.01$ is the significant level for Wald tests, and $\rho = 1$ is the percentage of the significant SNPs.

h	m	n	0%	10%	20%	30%	40%	50%
0.01	10	60	0.5008	0.5000	0.4990	0.5023	0.4995	0.5018
0.01	10	200	0.5008	0.4995	0.5002	0.5006	0.5018	0.5003
0.01	50	60	0.5045	0.5030	0.5018	0.4997	0.5000	0.4977
0.01	50	200	0.5011	0.5036	0.5029	0.5040	0.5012	0.4997
0.01	200	60	0.5073	0.5090	0.4993	0.5002	0.4977	0.4982
0.01	200	200	0.5020	0.5137	0.5052	0.5039	0.5042	0.5060
0.05	10	60	0.5117	0.5055	0.5072	0.5045	0.5083	0.5048
0.05	10	200	0.5128	0.5264	0.5228	0.5173	0.5158	0.5175
0.05	50	60	0.5250	0.5232	0.5300	0.5277	0.5283	0.5195
0.05	50	200	0.5494	0.5567	0.5517	0.5428	0.5473	0.5419
0.05	200	60	0.5642	0.5505	0.5552	0.5615	0.5497	0.5443
0.05	200	200	0.6113	0.6069	0.6076	0.6048	0.6035	0.5941
0.2	10	60	0.7795	0.7695	0.7517	0.7635	0.7278	0.7273
0.2	10	200	0.8670	0.8480	0.8264	0.8068	0.7863	0.7656
0.2	50	60	0.9558	0.9455	0.9398	0.9207	0.8993	0.8803
0.2	50	200	0.9914	0.9868	0.9789	0.9670	0.9519	0.9292
0.2	200	60	0.9773	0.9800	0.9795	0.9708	0.9747	0.9678
0.2	200	200	0.9992	0.9992	0.9988	0.9983	0.9964	0.9956

2.6 References

[1] Billingsely,P. (1979) Probability and Measure. Wiley Series in Probability and Mathematical Statistics, New York.

[2] Davies,R.W., Dandona,S., Stewart,A.F.R., Chen,L., Ellis,S.G., Tang,W.H.W., Hazen,S.L., Roberts,R., McPherson,R., and Wells,G.A. (2010) Improved prediction of cardiovascular disease based on a panel of single nucleotide polymorphisms identified through genome-wide association studies. *Circ Cardiovasc Genet* **3**, 468-474.

[3] De Valpine,P., Bitter,H.M., Brown,M.P.S., and Heller,J. (2009) A simulation-approximation approach to sample size planning for high-dimensional classification studies. *Biostatistics*, **10**, 424-435.

[4] Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D., Weingessel, A. (2005). e1071: Misc Functions of the Department of Statistics (e1071). *TU Wien*, **1**, 5-11. `http://CRAN.R-project.org/`.

[5] Dobbin,K.K. and Simon,R.M. (2005) Sample size determination in microarray experiments for class comparison and prognostic classification. *Biostatistics*, **6**, 27-38.

[6] Dobbin,K.K. and Simon,R.M. (2007) Sample size planning for developing classifiers using high-dimensional DNA microarray data. *Biostatistics*, **8**, 101-117.

[7] Dobbin,K.K., Zhao,Y., and Simon,R.M. (2008) How large a training set is needed to develop a classifier for microarray data. *Clin Cancer Res*, **14**, 108-114.

[8] Fu,W.J., Dougherty,E.R., Mallick,B., and Carroll,R.J. (2005) How many samples are needed to build a classifier: a general sequential approach. *Bioinformatics*, **21**, 63-70.

[9] Guzzetta,G., Jurman,G., and Furlanello,C. (2010). A machine learning pipeline for quantitative phenotype prediction from genotype data. *BMC Bioinformatics*, **11**(Suppl 8), S3.

[10] Kuznetsov,I.B., McDuffie,M., and Moslehi,R. (2009) A web server for inferring the human N-acetyltransferase-2 (NAT2) enzymatic phenotype from NAT2 genotype. *Bioinformatics*, **25**, 1185-1186.

[11] Lachenbruch,P.A. (1968). On expected probabilities of misclassification in discriminant analysis, necessary sample size, and a relation with the multiple correlation coefficient. *Biometrics*, **24**, 823-834.

[12] Lee,S.H., van der Werf,J.H.J., Hayes,B.J., Goddard,M.E., and Visscher,P.M. (2008) Predicting Unobserved Phenotypes for Complex Traits from Whole-Genome SNP Data. *Plos Genet*, **4**, e1000231.

[13] Mukherjee,S., Tamayo,P., Rogers,S., Rifkin,R., Engle,A., Campbell,C., Golub,T.R., and Mesirov,J.P. (2003) Estimating dataset size requirements for classifying DNA microarray data. *J Comput Biol*, **10**, 119-142.

[14] Nunkesser,R., Bernholt,T., Schwender,H., Ickstadt,K., and Wegener,I. (2007) Detecting high-order interactions of single nucleotide polymorphisms using genetic programming. *Bioinformatics*, **23**, 3280-3288.

[15] Onuki,R., Shibuya,T., and Kanehisa,M. (2010) New kernel methods for phenotype prediction from genotype data. *Genome Inform*, **22**, 132-141.

[16] Sabbagh,A. and Darlu,P. (2006) SNP selection at the NAT2 locus for an accurate prediction of the acetylation phenotype. *Genet Med*, **8**, 76-85.

[17] Wang,D.C. and Larder,B. (2003) Enhanced prediction of lopinavir resistance from genotype by use of artificial neural networks. *J Infect Dis*, **188**, 653-660.

[18] Wray,N.R., Goddard,M.E., and Visscher.P.M. (2007) Prediction of individual genetic risk to disease from genome-wide association studies. *Genome Res.*, **17**, 1520-1528.

[19] Zhou,N. and Wang,L. (2007) Effective selection of informative SNPs and classification on the HapMap genotype data. *BMC Bioinformatics*, **8**, 484.

Chapter 3

Determination of sample size for a multi-class classifier based on single-nucleotide polymorphisms: A Volume Under the Surface approach[2]

[2]Liu,X., Wang, Y. and Sriram, T. N. Submitted to: *Bioinformatics,*

Abstract

Data on single-nucleotide polymorphism (SNP) have been found to be useful in predicting phenotypes ranging from an individual's class membership to his/her risk of developing a disease. In multi-class classification scenarios, clinical samples are often limited by cost constraints, making it necessary to determine an adequate sample size needed to build a classifier based on SNPs. The performance of such classifiers can be assessed using the Area Under the Receiver Operating Characteristic (ROC) Curve (AUC) for two classes and the Volume Under the ROC hyper-Surface (VUS) for three or more classes. Sample size determination made based on AUC or VUS would not only guarantee an overall correct classification rate, but also make the studies more cost-effective.

For coded SNP data from $D(\geq 2)$ classes, we derive an optimal Bayes classifier and a linear classifier, and obtain a normal approximation to the probability of correct classification for each classifier. These approximations are used to evaluate the associated $AUCs$ or $VUSs$, whose performances are then validated via Monte Carlo simulations. We give an algorithm for sample size determination, which ensures that the difference between the two approximate $AUCs$ (or $VUSs$) is below a pre-specified threshold. The performance of this algorithm is also illustrated via simulations. For the *HapMap* data with three and four populations, a linear classifier is built using 92 independent SNPs and the required total sample sizes are determined for various threshold values. We also illustrate the usefulness of our sample size determination algorithm in a prediction problem using a *Heterogeneous Stock Mice* data, where the continuous variable *Anxiety* is categorized into three groups, whereas the variable *Obesity BMI* is categorized into four groups, and then a linear classifier is built based on 348 SNPs for each variable.

Key words and Phrases: Area Under the Receiver Operating Characteristic Curve; Classification; Hapmap data; Heterogeneous Stock Mice data; Probability of correct classification; Receiver Operating Characteristic; Sample Size Determination;

Single-nucleotide polymorphisms; Volume Under the Receiver Operating Characteristic hyper-Surface; Wald test.

3.1 Introduction

Data on single-nucleotide polymorphisms (SNPs) have been found to be useful in predicting an individual's class membership or his/her response to a drug, susceptibility to environmental factors such as toxins, and risk of developing a particular disease, among others (Guzzetta, Jurman and Furlanello, 2010; Lee *and others*, 2008; Nunkesser *and others*, 2007; Wray, Goddard and Visscher, 2007; Zhou, 2007). In the classification literature, there are a variety of classifiers (e.g., Support Vector Machine, genetic programming, Neural Networks and Logistic Regression) and sample size determination methods (De Valpine *and others*, 2009; Dobbin and Simon, 2005, 2007; Dobbin, Zhao and Simon, 2008; and Mukherjee *and others*, 2003), but the latter methods are mostly applicable for continuous data.

For coded SNP data from two classes, recently Liu et al. (2012) developed an optimal Bayes classifier and a linear classifier, and obtained a normal approximation to the probability of correct classification (PCC) for each classifier. They also proposed an algorithm to determine an adequate sample size, which ensures that the difference between the two approximate $PCCs$ is below a pre-specified threshold value. Via Monte Carlo simulations, Liu et al. (2012) assessed the performance of their approximations and illustrated the performance of their sample size determination algorithm using simulations and an application to Hapmap data on Chinese and Japanese populations.

While Liu et al. (2012)'s sample size determination criterion based on approximate $PCCs$ performed well, they remarked (see their REMARK1) that the choice of their discrimination value for their classifiers requires further maximization, which places additional an computational burden. Such choices of discrimination values become even more complicated, when there are three or more classes. A well known way to overcome this problem in a two-class scenario is to consider the Receiver Operating Characteristic (ROC) curve, which plots the

True Positive Rates vs. False Positives Rates, at various discrimination values (Metz, 1978; Fawcett, 2005). Note that the ROC allows the discrimination value to be varied, and explores all possible combinations of the correct classification rates (Landgrebe and Duin, 2007). The Area Under the ROC curve (AUC) is commonly used as a scalar performance measure, which allows classifiers to be compared independent of discrimination values. However, the AUC is only applicable in a two-class case. A popular multi-class extension of the AUC measure is known as the Volume Under the ROC hyper-Surface (VUS) (see e.g., Landgrebe and Duin, 2007 and Landgrebe and Paclik, 2010).

This article revisits the problem of sample size determination, when a coded SNP data is observed, but uses AUC and VUS as performance measures for two-class and multi-class cases, respectively. More specifically, for a coded SNP data from $D(\geq 2)$ classes, in Section 3.2.2 we derive an optimal Bayes classifier and obtain a normal approximation to the probability of correct classification, denoted by $PCC(\infty)$, based on the assumptions made in Section 3.2.1. In Section 3.2.3, We derive a linear classifier and, once again, obtain a normal approximation the probability of correct classification, denoted by $PCC(\vec{n})$. For an overall assessment of each of the classifiers, in Section 3.2.4 we define the scalar measures, AUC for two-class and VUS for multi-class, whereas the computations of $AUC(\infty), AUC(\vec{n}), VUS(\infty)$ and $VUS(\vec{n})$ are described in Section 3.2.5. In Section 3.2.6, for the two-class case, we propose to determine the sample size n for which $AUC(\infty) - AUC(\vec{n}) < \gamma$, or $AUC(\vec{n}) > \gamma'$ for a pre-specified threshold value γ or γ'. Similarly, for the multi-class case, we propose to determine the sample size n for which $VUS(\infty) - VUS(\vec{n}) < \gamma$, or $VUS(\vec{n}) > \gamma'$ for a pre-specified threshold value γ or γ'. A computational algorithm to determine the sample sizes for various values of γ (or γ') is given in Section 3.3.1. While Monte Carlo simulations given in Section 3.3.1 corroborate our theory, the performance of sample size determination algorithm is assessed by applying it to the *HapMap* data consisting of 3 and 4 populations, respectively.

Whereas classification of an observation into one of D classes based on a coded SNP data is important, predicting the value of a future phenotype based on SNP data are also of interest. Lee *and others*, 2008 predicted three traits of stock mouse, including coat color (measure of darkness), percentage of $CD8+$ cells and mean cellular haemoglobin, using an additive genetic model. Wray *and others*, 2007 built a model to predict the risk of a disease. Wei *and others*, 2009 predicted the risk of type 1 diabetes using the support vector machine. Lorenzana and Bernarodo (2009) predicted plant traits, such as, plant height, ear height, root lodging, and so on, using multiple linear regression and best linear unbiased prediction. De Roos *and others* (2009) built a model to predict the breeding value of bulls. While there is an abundance of literature on prediction, there is no research on the determination of sample size required for prediction based on SNPs. For quantitative trait prediction, we can divide the values of a trait into several classes possibly of equal width, turning it into a multi-class classification problem. In fact, in Section 3.3.2, we illustrate the usefulness of our sample size determination algorithm described above in a prediction problem using a *Heterogeneous Stock Mice* data, where the continuous variable, *Anxiety*, is categorized into 3 classes using two cut-off points based on its distribution and *Obesity BMI* is categorized into four classes using its quartiles, and then a linear classifier is built based on 348 SNPs. The R code for our computational algorithm is available under journal's supplementary information site.

3.2 Methods

3.2.1 Assumptions

Suppose there are $D(\geq 2)$ distinct classes denoted by $C_1, \ldots, C_D$, consisting of $n_1, \ldots, n_D$ subjects, respectively. The basic assumptions made below are same as those in Liu et al. (2012), but we list them for completeness sake. For each subject, we observe a p-dimensional SNP vector, $\vec{x} = (x_1, x_2, .., x_p)'$, where typically p is much larger ($>>$) than $\sum_{i=1}^{D} n_i$, and the jth SNP is coded in such a way $x_j = 0, 1, 2$, which denotes the number of minor alleles in the genotype "aa", "Aa" and "AA", respectively. It is possible that some of the SNPs are

highly correlated, leading us to choose one SNP to represent a set of highly correlated ones. For classification and sample size determination, the following assumptions hold throughout the article:

1. The data vector is, $\vec{x} = (x_1, \ldots, x_m)'$, consisting only of $m >> n_1 + n_2 + ... + n_D$ statistically independent SNPs; the rest of the $(p - m)$ SNPs are not used for classification.

2. For each $k = 1, \ldots, D$ and $j = 1, \ldots, m$, we postulate the Hardy-Weinberg equilibrium, according to which the probability mass function of the coded SNP (X_j) belonging to class k is given by

$$P_k(X_j = x_j|\theta_{k,j}) = \binom{2}{x_j}\theta_{k,j}^{x_j}(1-\theta_{k,j})^{2-x_j}, \quad x_j = 0, 1, 2,$$

where, by the definition of SNP, $\theta_{k,j} \in (0.01, 0.5)$. Here, $\theta_{k,j} < 0.5$ because it is the minor allele frequency and $\theta_{k,j} > 0.01$ ensures that the polymorphism is not a *mutation*. For each $k = 1, \ldots, D$, let $\vec{\theta}_k = (\theta_{k,1}, \ldots, \theta_{k,m})'$ denote the parameter vector corresponding to the class C_k.

3. There is a percentage ρ of the m SNPs with marginal effect on any two classes. Let $l = \lfloor \rho m \rfloor$ SNPs with marginal effects. Furthermore, for any $(D-1) \times 1$ real vector $\vec{\boldsymbol{\beta}} \neq \vec{\mathbf{0}}$ assume that $\sum_{j=1}^{l}\left[\sum_{k'=1,k'\neq k}^{D}\beta_{k'}\log(\frac{\theta_{k,j}(1-\theta_{k',j})}{\theta_{k',j}(1-\theta_{k,j})})\right]^2 \to \infty$, as $l \to \infty$.

3.2.2 The optimal classifier and its PCC

By the assumptions above, the conditional mass function of $\vec{X} = (X_1, ..., X_l)'$ given the class C_k, $k = 1, \ldots, D$, is

$$f_k(\vec{X} = \vec{x}|\vec{\theta}_k) = \prod_{j=1}^{l}\{\binom{2}{x_j}\theta_{k,j}^{x_j}(1-\theta_{k,j})^{2-x_j}\}.$$

For each $1 \leq k \leq D$ and $f(\vec{x}) = \sum_{k=1}^{D}\pi_k f_k(\vec{x}|\vec{\theta}_k)$, the posterior mass function of the class C_k given $\vec{x}$ is

$$\tau_k(\vec{\theta}_k|\vec{x}) = \frac{\pi_k f_k(\vec{x}|\vec{\theta}_k)}{f(\vec{x})}.$$

For any fixed $k = 1, \ldots, D$, the Bayes classification rule then classifies $\vec{x}$ to C_k if

$$\frac{\tau_k(\vec{\theta}_k|\vec{x})}{\tau_{k'}(\vec{\theta}_{k'}|\vec{x})} > 1 \tag{3.1}$$

for all $k' \neq k$, which, after some algebra, is

$$\log(\frac{\tau_k(\vec{\theta}_k|\vec{x})}{\tau_{k'}(\vec{\theta}_{k'}|\vec{x})}) = \sum_{j=1}^{l} b_{k,k'}^j x_j - K_{k,k'} > 0, \tag{3.2}$$

where

$$\begin{aligned} b_{k,k'}^j &= \log(\frac{\theta_{k,j}(1-\theta_{k',j})}{\theta_{k',j}(1-\theta_{k,j})}) \\ K_{k,k'} &= \log(\frac{\pi_{k'}}{\pi_k}) + 2\log(\frac{1-\theta_{k',j}}{1-\theta_{k,j}}). \end{aligned} \tag{3.3}$$

Then, (3.2) leads to the *optimal* classifier, which classifies $\vec{x}$ to C_k if

$$\sum_{j=1}^{l} b_{k,k'}^j x_j > K_{k,k'} \tag{3.4}$$

for all $k' \neq k$. Then, the PCC of the optimal classifier is defined as

$$PCC(\infty) = \sum_{k=1}^{D} \pi_k P(\bigcap_{k' \neq k} \{\sum_{j=1}^{l} b_{k,k'}^j x_j > K_{k,k'}\} | \vec{X} \in C_k).$$

To obtain an approximation for $PCC(\infty)$, fix k $(= 1, \ldots, D)$ and consider the $(D-1)\times 1$ vector $\vec{\mathbf{Y}}_{l,k}$ with components $Y_{l,k}(k') = \sum_{j=1}^{l} b_{k,k'}^j x_j$, for $k' \neq k$. Then, by the assumptions in Section 3.2.1, the mean of $\vec{\mathbf{Y}}_{l,k}$, $\vec{\boldsymbol{\mu}}_{l,k} = E(\vec{\mathbf{Y}}_{l,k})$, is a $(D-1)\times 1$ vector with components $\mu_{l,k}(k') = \sum_{j=1}^{l} 2\theta_{k,j} b_{k,k'}^j$, and the Covariance of $\vec{\mathbf{Y}}_{l,k}$, $\boldsymbol{\Sigma}_{l,k} = Cov(\vec{\mathbf{Y}}_{l,k})$, is a $(D-1)\times(D-1)$ matrix with its (k', k'')-th element, $Cov(Y_{l,k}(k'), Y_{l,k}(k'')) = \sum_{j=1}^{l} 2\theta_{k,j}(1-\theta_{k,j}) b_{k,k'}^j b_{k,k''}^j$ for $k \neq k', k''$.

Then, for any $(D-1)\times 1$ real vector $\vec{\boldsymbol{\beta}}$ and for each k, ,

$$\frac{\vec{\boldsymbol{\beta}}\,'\vec{\mathbf{Y}}_{l,k} - \vec{\boldsymbol{\beta}}\,'\vec{\boldsymbol{\mu}}_{l,k}}{\sqrt{\vec{\boldsymbol{\beta}}\,'\boldsymbol{\Sigma}_{l,k}\vec{\boldsymbol{\beta}}}} \Rightarrow N(0,1) \quad \text{as} \quad l \to \infty;$$

see Appendix 1 for details. Then, by the Cramér-Wold device (see Billingsley, 1995, Theorem 29.4) we have that, for large l

$$\vec{\mathbf{Y}}_{l,k} \approx \mathbf{N}(\vec{\boldsymbol{\mu}}_{l,k}, \boldsymbol{\Sigma}_{l,k}), \tag{3.5}$$

where $\mathbf{N}$ denotes the $(D-1)$-dimensional multivariate normal distribution.

Note from (3.3) that $K_{k',k''} = -K_{k'',k'}$ and $\sum_{k=s}^{t} K_{k,(k+1)} = K_{s,t+1}$ for any $s, t = 1, \ldots, D-1$ with $s \leq t$. Let $K_{i,i+1} = K_i$ for $i = 1, ..., D-1$ and define

$$\vec{\mathbf{K}}_1 = \begin{pmatrix} K_1 \\ K_1 + K_2 \\ ... \\ K_1 + K_2 + ... + K_{D-1} \end{pmatrix}_{(D-1)\times 1} \tag{3.6}$$

Then, for $k = 2, ..., D$, it can be shown that

$$\vec{\mathbf{K}}_k = \vec{\mathbf{K}}_{(k-1)} - K_{(k-1)}\vec{\mathbf{1}} - K_{(k-1)}\vec{\mathbf{e}}_{(k-1)} \tag{3.7}$$

where $\vec{\mathbf{1}} = (1, 1, ..., 1)'$ and $\vec{\mathbf{e}}_{k-1} = (0, ..., 0, 1, 0, ..., 0)'$ with 1 in the $(k-1)$-th position and 0 elsewhere. Now, for any $(D-1) \times 1$ vector $\vec{\mathbf{K}}$, define

$$\tilde{\Phi}(\vec{\mathbf{K}}; \vec{\boldsymbol{\mu}}, \boldsymbol{\Sigma}) = \int_{\vec{\mathbf{K}}}^{\infty} \boldsymbol{\phi}(\vec{\mathbf{x}}; \vec{\boldsymbol{\mu}}, \boldsymbol{\Sigma}) d\vec{\mathbf{x}}, \tag{3.8}$$

where $\boldsymbol{\phi}$ is the $(D-1)$-dimensional multivariate normal density and $\int_{\vec{\mathbf{K}}}^{\infty}$ is a multiple integral. Then, by (3.5) and (3.8), for large l

$$\begin{aligned} PCC(\infty) &= \sum_{k=1}^{D} \pi_k P(\vec{\mathbf{Y}}_{l,k} > \vec{\mathbf{K}}_k | \vec{X} \in C_k) \\ &\approx \sum_{k=1}^{D} \pi_k \tilde{\Phi}(\vec{\mathbf{K}}_k; \vec{\boldsymbol{\mu}}_{l,k}, \boldsymbol{\Sigma}_{l,k}). \end{aligned} \tag{3.9}$$

We give an expression for (A.5) for the case $D = 3$ in Appendix 4.

3.2.3 A linear classifier and its PCC

The optimal classifier in (3.4) suggests considering the following linear classifier, which classifies $\vec{x}$ to C_k if

$$\sum_{j=1}^{m} \hat{b}^j_{k,k'} w_{j,n}(k,k') x_j > \tilde{K}_{k,k'} \tag{3.10}$$

for all $k' \neq k$, where $\hat{b}^j_{k,k'} = \log(\frac{\hat{\theta}_{k,j}(1-\hat{\theta}_{k',j})}{\hat{\theta}_{k',j}(1-\hat{\theta}_{k,j})})$, $\hat{\theta}_{k,j}$ and $\hat{\theta}_{k',j}$ are the maximum likelihood estimators of $\theta_{k,j}$ and $\theta_{k',j}$, respectively, and for each $j = 1, \ldots, m$, if the SNP x_j is determined to have a marginal effect on $C_k \& C_{k'}$ by a test of hypothesis $H^{k,k'}_{0,j} : \theta_{k,j} = \theta_{k',j}$ versus $H^{k,k'}_{1,j} : \theta_{k,j} \neq \theta_{k',j}$, then $w_{j,n}(k,k') = 1$ if $H^{k,k'}_{0,j}$ is rejected; else $w_{j,n}(k,k') = 0$. In Appendix 2, we use large sample theory to derive a Wald test of level α to test $H^{k,k'}_{0,j}$ versus $H^{k,k'}_{1,j}$, and an expression for the power, $1 - \beta^{k,k'}_j(n_k, n_{k'}, h_j)$, of this test, when $\theta_{k,j} - \theta_{k',j} = h_j$. The power, $1 - \beta^{k,k'}_j(n_k, n_{k'}, h_j)$, of the test is determined using a non-central Chi-square distribution with a non-centrality parameter, which depends on $n_k + n_{k'}$ and h_j; see Appendix 2 for details.

Then, it can be shown that

$$\begin{aligned} P(Reject\ H^{k,k'}_{0,j}) &= P(Reject\ H^{k,k'}_{0,j} | H^{k,k'}_{0,j}) P(H^{k,k'}_{0,j}) \\ &\quad + P(Reject\ H^{k,k'}_{0,j} | H^{k,k'}_{1,j}) P(H^{k,k'}_{1,j}) \\ &= [\rho\{1 - \beta^{k,k'}_j(n_k, n_{k'}, h)\} + (1-\rho)\alpha] = \tilde{\eta}^{k,k'}_j, \end{aligned} \tag{3.11}$$

where ρ is from Assumption 3 in Section 3.2.1 and $\tilde{\eta}^{k,k'}_j$ depends on $(n_k, n_{k'})$. Therefore, from the definition of $w_{j,n}(k,k')$ above,

$$E(w_{j,n}(k,k')) = E((w_{j,n}(k,k'))^2) = P(Reject\ H^{k,k'}_{0,j}) = \tilde{\eta}^{k,k'}_j.$$

As in Section 3.2.2, consider the $(D-1) \times 1$ vector $\tilde{\vec{\mathbf{Y}}}_{n,m,k}$ with components $\tilde{Y}_{n,m,k}(k') = \sum_{j=1}^m \hat{b}^j_{k,k'} w_{j,n}(k,k') x_j$, for $k' \neq k$. Then, by the assumptions in Section 3.2.1 and that $(\hat{b}^j_{k,k'} - b^j_{k,k'}) = O(n^{-1/2})$, it is shown in Appendix 3 that the mean, $E(\tilde{\vec{\mathbf{Y}}}_{n,m,k}) = \tilde{\vec{\boldsymbol{\mu}}}_{m,k}$, is a $(D-1) \times 1$ vector with components $\tilde{\vec{\mu}}_{m,k}(k') \approx \sum_{j=1}^m 2\theta_{k,j} b^j_{k,k'} \tilde{\eta}^{k,k'}_j$. Similarly, we also compute an

approximate expression for the $(D-1)\times(D-1)$ Covariance matrix of $\tilde{\vec{\mathbf{Y}}}_{n,m,k}$, denoted by $\tilde{\mathbf{\Sigma}}_{m,k}$. See Appendix 3 for these details.

For simplicity, assume that $\tilde{K}_{k,k'}$ in (3.10) also satisfies the same properties as $K_{k,k'}$ in (3.4); that is, $\tilde{K}_{k',k''} = -\tilde{K}_{k'',k'}$ and $\sum_{k=s}^{t}\tilde{K}_{k,(k+1)} = \tilde{K}_{s,t+1}$ for any $s,t = 1,\ldots,D-1$ with $s \le t$. Let $\tilde{K}_{i,i+1} = \tilde{K}_i$ for $i = 1,...,D-1$, and define $\tilde{\vec{\mathbf{K}}}_1$ and $\tilde{\vec{\mathbf{K}}}_k$ as in (3.6) and (3.7), respectively. Then, once again, as in (3.5), for large m we can show that $\tilde{\vec{\mathbf{Y}}}_{n,m,k} \approx \mathbf{N}(\tilde{\vec{\boldsymbol{\mu}}}_{m,k}, \tilde{\mathbf{\Sigma}}_{m,k})$.

Then, as in (A.5), the PCC of the linear classifier and its approximation are given by

$$\begin{aligned} PCC(\vec{n}) &= \sum_{k=1}^{D}\pi_k P(\tilde{\vec{\mathbf{Y}}}_{n,m,k} > \tilde{\vec{\mathbf{K}}}_k | \vec{X} \in C_k) \\ &\approx \sum_{k=1}^{D}\pi_k \tilde{\Phi}(\tilde{\vec{\mathbf{K}}}_k; \tilde{\vec{\boldsymbol{\mu}}}_{m,k}, \tilde{\mathbf{\Sigma}}_{m,k}). \end{aligned} \tag{3.12}$$

Note that $PCC(\vec{n})$ depends on $\vec{n} = (n_1,\ldots,n_D)'$ through $(\tilde{\vec{\boldsymbol{\mu}}}_{m,k}, \tilde{\mathbf{\Sigma}}_{m,k})$, which depend on $\{\tilde{\eta}_j^{k,k'}; k,k' = 1,\ldots,D\}$; see Appendix 3. We give an expression for (3.12) for the case $D = 3$ in Appendix 4.

3.2.4 *AUC & VUS* for the optimal and linear classifiers

For any (k,k'), define

$$\xi_{k,k'} = P(Classify\ \vec{X}\ to\, C_{k'} | \vec{X} \in C_k).$$

Then, for the optimal classifier in (3.4) we have from (A.5) that

$$\xi_{k,k} = P(\vec{\mathbf{Y}}_{l,k} > \vec{\mathbf{K}}_k | \vec{X} \in C_k) \approx \tilde{\Phi}(\vec{\mathbf{K}}_k; \vec{\boldsymbol{\mu}}_{l,k}, \mathbf{\Sigma}_{l,k}) \tag{3.13}$$

and for the linear classifier in (3.10) we have from (3.12) that

$$\tilde{\xi}_{k,k} = P(\tilde{\vec{\mathbf{Y}}}_{n,m,k} > \tilde{\vec{\mathbf{K}}}_k | \vec{X} \in C_k) \approx \tilde{\Phi}(\tilde{\vec{\mathbf{K}}}_k; \tilde{\vec{\boldsymbol{\mu}}}_{m,k}, \tilde{\mathbf{\Sigma}}_{m,k}),$$

for $k = 1,\ldots,D$. When $D = 2$, the $ROC(\infty)$ for two classes is the curve of $\xi_{2,2}$ vs. $(1-\xi_{1,1})$ for the optimal classifier. Then, the $AUC(\infty)$ is

$$AUC(\infty) = \int \xi_{2,2} d\xi_{1,1}.$$

However, when the number of classes $D \geq 3$, we need to consider the volume under the *ROC* hypersurface. Following the work of Landgrebe and Duin (2007), the VUS is defined as

$$VUS(\infty) = \int \dots \int \xi_{D,D} d\xi_{1,1}\xi_{2,2}...\xi_{D-1,D-1} \tag{3.14}$$

$$= \int ... \int \xi_{D,D} |\frac{\partial(\xi_{1,1}, \xi_{2,2}, ..., \xi_{D-1,D-1})}{\partial(K_1, K_2, ..., K_{D-1})}| dK_1...dK_{D-1}.$$

We replace $\xi_{k.k}$ by $\tilde{\xi}_{k,k}$ in the above definitions of ROC, AUC and the VUS for the linear classifier in (3.10), and we denote the resulting ones as $AUC(\vec{n})$ and $VUS(\vec{n})$. We give an expression for (3.13) and (3.14) for the case $D = 3$ in Appendix 4.

3.2.5 Computation method of VUS

Here, we describe the computation of the high dimensional integration in $VUS(\infty)$; see (3.14). We will denote $\xi_k = \xi_{k,k}, k = 1, \dots, D$ for this discussion. First, we randomly generate the thresholds $\vec{K} = (K_1, K_2, ..., K_{D-1})$ (see (3.6)) and compute the corresponding $\vec{\xi} = (\xi_1, \xi_2, ..., \xi_D)'$ satisfying (3.13). Note that the $\vec{\xi}$ contributes to the integration in VUS, only if all the ξ_k's are positive. To find as many $\vec{\xi}$'s that contribute to the integration, we use the *ant colony* optimization algorithm, where only the $\vec{K}$'s corresponding to the $\vec{\xi}$'s that contribute to the integration are retained, but these are perturbed by a small noise and the resulting K's are used as seed for the next iteration. Then, we use the genetic algorithm to obtain another set of $\vec{\xi}$ located in a different region in $(0, 1)^k$ that also contribute to the integration. We use the *ant colony* algorithm and the genetic algorithm alternatively to eventually generate a dense set of $\vec{\xi} \in (0, 1)^k$ that contribute to the integration. Note that the process is such that the newly generated $\vec{\xi}$'s are appended to all the previously generated $\vec{\xi}$'s. To compute the volume, $VUS(\infty)$, we use the *convhulln* function in *qhull* R-package. However, the *convhulln* function can find the convex hull of the D-dimension points and compute the volume of the convex hull. Therefore, to compute the volume (under the $\vec{\xi}$'s), $VUS(\infty)$, a base of $\vec{\xi}$ (same as $\vec{\xi}$ except that one of the components, e.g. the first component, is set 0) is appended to original $\vec{\xi}$. Since in each iteration the new $\vec{\xi}$'s are appended to the old $\vec{\xi}$ from the

previous iterations, and the VUS is concave, the computed VUS is supposed increase with each iteration. We stop appending new $\vec{\xi}$'s when $|VUS(old) - VUS(new)| < 0.001$ and when this criterion is satisfied, we get the value of $VUS(\infty)$. The $AUC(\infty), AUC(\vec{n}), VUS(\vec{n})$ are computed in a similar way.

3.2.6 Sample Size determination using VUS or AUC

Given a threshold γ, we determine the sample size is n satisfying the following condition:

$$VUS(\infty) - VUS(\vec{n}) < \gamma \tag{3.15}$$

We can also define an alternative criterion, which determines n satisfying $VUS(\vec{n}) > \gamma'$, for a γ'. For $D = 2$ case, an analogous sample size determination criteria can be defined using $AUC(\infty)$ and $AUC(\vec{n})$. A simulation for the case $D = 2$ is carried out in Appendix 5 to study the performance of our sample size determination algorithm.

3.3 Numerical Results

3.3.1 Monte Carlo simulations

Before we illustrate the performance of our sample size determination algorithm based on AUC or VUS, we present below results from elaborate Monte Carlo simulations conducted to verify the accuracy of the approximation for $AUC(\vec{n})$ and $VUS(\vec{n})$, and study its behavior as a function of n and other parameters. Here, we present numerical assessments based on VUS for 3 and 4 classes, but those based on AUC for two classes are given in Appendix 5. Henceforth, we will set $n_k = n$ for all $k = 1, \ldots, D$, and we will use n instead of $\vec{n}$ to simplify notations.

When $D = 3$, we consider the following simulation set up: For $\vec{\theta}_1 = (\theta_{1,1}, \ldots, \theta_{1,m})'$, let $\theta_{1,j} \sim U(0.4, 0.49), j = 1, \ldots, m$; for a specified scalar value h, let $\vec{h}_1, \vec{h}_2$ be such that its components $h_{i,j} \sim U(h-0.002, h+0.002), i = 1, 2; j = 1, \ldots, m$; and let $\vec{\theta}_2 = \vec{\theta}_1 - \vec{h}_1, \vec{\theta}_3 = \vec{\theta}_2 -$

$\vec{h}_2$. First, we generate a $(\vec{\theta}_1, \vec{\theta}_2, \vec{\theta}_3)$ according to the above set up, and then generate the data $\vec{x}$ for each class. We then compute $\widehat{VUS(\infty)}$ and $\widehat{VUS(n)}$, which are based on approximations in (A.5) and (3.12), respectively. For this $(\vec{\theta}_1, \vec{\theta}_2, \vec{\theta}_3)$, we then draw twenty $\vec{x}$ data sets , and calculate a Monte Carlo estimate, denoted by $\widehat{VUS(n)}$MC. This process was repeated 20 times in order to compute the average correct classification frequency, $\widehat{VUS(n)}$MC, given in Table 3.1. It is evident from Table 3.1 that the $\widehat{Bias} = \widehat{VUS(n)}\text{MC} - \widehat{VUS(n)}$ is negligible in most cases, thereby validating the use of our approximation for $VUS(n)$. Table 3.1 also gives similar results for the case $D = 4$. Note that for a random classifier, the $VUS(\infty) = 1/D!$, which is the lower bound of $VUS(\infty)$ for any classifer.

Table 3.1: Performance of Optimal and Linear classifiers. Here, $D = 3$ and 4, $\vec{\theta}_1 = (\theta_{1,1}, \ldots, \theta_{1,m})'$, let $\theta_{1,j} \sim U(0.4, 0.49), j = 1, \ldots, m$; for a specified scalar value h, let $\vec{h}_1, \vec{h}_2, \vec{h}_3$ be such that its components $h_{i,j} \sim U(h - 0.002, h + 0.002), j = 1, \ldots, m$; and let $\vec{\theta}_{i+1} = \vec{\theta}_i - \vec{h}_i, i = 1, 2, 3$; n is the sample size for each class; m is the number of independent SNPs, $\alpha = 0.01$ is the significant level for Wald tests; and $\rho = 1$ is the percentage of the significant SNPs.

			$D = 3$			
h	m	n	$VUS(\infty)$	$VUS(n)$	$VUS(n)MC$	Bias
0.02	50	50	0.3013	0.1772	0.1657	-0.0116
0.02	50	100	0.3015	0.1793	0.1742	-0.0052
0.02	100	50	0.3662	0.1807	0.1874	0.0067
0.02	100	100	0.366	0.1837	0.1974	0.0136
0.05	50	50	0.5469	0.2229	0.2442	0.0213
0.05	50	100	0.5467	0.2517	0.2845	0.0328
0.05	100	50	0.6988	0.2448	0.2912	0.0463
0.05	100	100	0.6987	0.2848	0.3377	0.0529
0.1	50	50	0.8686	0.4179	0.4675	0.0496
0.1	50	100	0.8687	0.4958	0.55	0.0542
0.1	100	50	0.9667	0.4776	0.5342	0.0566
0.1	100	100	0.9667	0.5692	0.6341	0.0649
			$D = 4$			
h	m	n	$VUS(\infty)$	$VUS(n)$	$VUS(n)MC$	Bias
0.02	50	50	0.1319	0.048	0.0462	-0.0018
0.02	50	100	0.1318	0.05	0.0512	0.0013
0.02	100	50	0.1892	0.0503	0.057	0.0068
0.02	100	100	0.189	0.0531	0.0614	0.0082
0.05	50	50	0.3891	0.0893	0.0923	0.003
0.05	50	100	0.3893	0.1175	0.1144	-0.0032
0.05	100	50	0.5832	0.1092	0.1127	0.0034
0.05	100	100	0.5831	0.1458	0.1285	-0.0174
0.1	50	50	0.8376	0.2933	0.2705	-0.0228
0.1	50	100	0.8378	0.4059	0.3517	-0.0542
0.1	100	50	0.9623	0.3653	0.3119	-0.0534
0.1	100	100	0.9626	0.4962	0.4085	-0.0877

Table 3.2: Sample size determination: Here, $D = 3$ and 4, and n is the sample size for each class satisfying: $\widehat{VUS(\infty)} - \widehat{VUS(n)} < \gamma \ (= 0.01)$. Here, $\vec{\theta}_1 = (\theta_{1,1}, \ldots, \theta_{1,m})'$, let $\theta_{1,j} \sim U(0.4, 0.49), j = 1, \ldots, m$; for a specified scalar value h, let $\vec{h}_1, \vec{h}_2, \vec{h}_3$ be such that its components $h_{i,j} \sim U(h - 0.002, h + 0.002), j = 1, \ldots, m$; and let $\vec{\theta}_{i+1} = \vec{\theta}_i - \vec{h}_i, i = 1, 2, 3$; m is the number of independent SNPs, $\alpha = 0.01$ is the significant level for Wald tests; and $\rho = 1$ is the percentage of the significant SNPs.

		n			
D	h	$m = 30$	$m = 50$	$m = 100$	$m = 200$
3	0.05	1957	2040	2091	2040
3	0.1	489	475	412	288
3	0.15	189	161	105	69
4	0.05	1923	2051	2137	2122
4	0.1	490	476	417	297

Next, we determine the smallest n such that $f(n) = \widehat{VUS(\infty)} - \widehat{VUS(n)} - \gamma < 0$, for a pre-specificed γ. We use the following algorithm to determine such an n: (i) Let $n = n_S$ and n_L such that $f(n_S) > 0$ and $f(n_L) < 0$, and set $n_M = [(n_S + n_L)/2]$. To begin the algorithm select a small n_S and a large n_L; (ii) If $f(n_M)f(n_S) < 0$, then reset $n_L = n_M$; else, reset $n_S = n_M$. In either case, return to step (i), unless $n_L - n_S \leq 1$, in which case, the smallest sample $n = n_L$; (iii) Use the smallest (total) sample of size Dn_L, with $n = n_L$ from each class, $C_1, \ldots, C_D$. We implemented this algorithm for each value of h, m and significance level α for the Wald test. Table 3.2 shows the determined sample sizes for each combination of parameters. From Table 3.2, it is evident that the required sample size reduces as h increases, as expected. Hence, $f(n) < 0$ for smaller sample sizes, as shown in Table 3.2. However, the effect of m on the determined sample sizes is less clear. When h is large, say $h \geq 0.1$, then the required sample size reduces as m becomes large. Whereas, when h is small, say $h = 0.05$, the reverse is true as m becomes large.

3.3.2 Application to real data

Application to the HapMap Data:

The aim of the International HapMap Project is to develop a haplotype map of the human genome, the so-called HapMap, which will describe the common patterns of human DNA sequence variation. Discovering the DNA sequence variants that contribute to common disease risk offers one of the best opportunities for understanding the complex causes of disease in humans (`http://hapmap.ncbi.nlm.nih.gov/abouthapmap.html`). The HapMap data (phase 3) consists of eleven populations with about $p = 1.2 \times 10^6$ SNPs. Here, we consider the following four classes: C_1 – the Han Chinese individuals from Beijing (CHB) with 137 subjects, C_2 – the Japanese individuals from Tokyo (JTP) with 113 subjects, C_3 – Chinese in Metropolitan Denver, Colorado (CHD) with 109 subjects, C_4 – Gujarati Indians in Houston, Texas (GIH) with 101 subjects.

Based on all the available subjects, we extracted pair-wise independent SNPs using the following procedure. Suppose L is a set of SNPs, then (I) form the set S with one SNP from L and update S after next step; (II) from the remaining SNPs in L, choose one SNP that is independent of every SNP in S using Kendall's τ coefficient as a test statistic to test pair-wise independence, and then add this new SNP to S. Here, we concluded independence if the Kendall's τ-value < 0.05; (III) Repeat (II) until each remaining SNP in L is correlated with at least one SNP in S. This procedure yielded a set S with $m = 92$ pair-wise independent SNPs, and with these we built our linear classifier.

Next, we set $\rho = 1$ so that $m = l = 92$. Recall that $\vec{\theta_k} = (\theta_{k,1}, \ldots, \theta_{k,l})'$ for $k = 1, \ldots, D$. We estimated $\vec{\theta_k}$ using the maximum likelihood (ML) estimates (based on 137, 113, 109 and 101 subjects, respectively). We substituted these ML estimates into the expressions for $\widehat{VUS(\infty)}$ and $\widehat{VUS(n)}$, respectively. Figure 3.1 shows plots of sample size required under different thresholds for 3 and 4 populations. From these figures, the sample size can be estimated. For example, if we set $\widehat{VUS(\infty)} - \widehat{VUS(n)} < 0.05$, then in three populations (CHB, JPT and CHD) case, about 500 samples are required for each class with a total

sample size of $1,500$, while in four populations (CHB, JPT, CHD and GIH) case, about 700 samples are required for each class with a total sample of 2800.

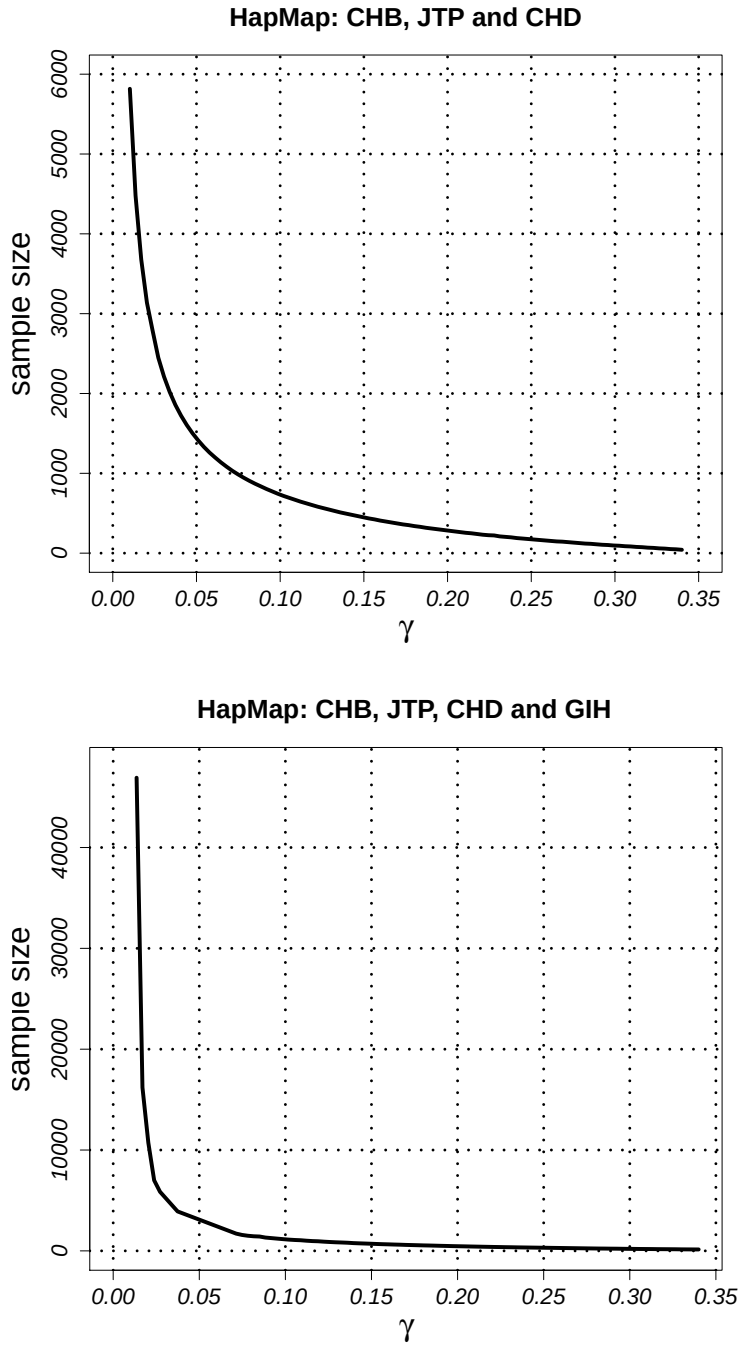

Figure 3.1: The sample size required for 3 and 4 populations in the *HapMap* data under different thresholds, γ. Here, $\rho = 1$, $\alpha = 0.1$, $m = 92$. The upper panel is for $D = 3$ populations (CHB, JTP and CHD), and $\widehat{VUS(\infty)} = 0.6178$. The lower panel is for $D = 4$ populations (CHB, JTP, CHD and GIH), and $\widehat{VUS(\infty)} = 0.5580$.

Application to the Heterogeneous Stock Mice Data:

We apply our multi-class based prediction method to a publicly available data, called *Stock Mice*, containing information about the pedigree, genotypes and phenotypes on heterogeneous stock mice (`http://gscan.well.ox.ad.uk/`). This data has 15,348 SNPs collected from 2,296 animals, belonging to 85 unrelated families, each having 17 traits. We determine sample sizes for the prediction of two of these 17 traits, namely, *anxiety* and *obesity BMI*, using our multi-class SNP based prediction method.

More specifically, the sample size determination method proposed above can also be used in prediction of a future phenotype. The idea is to categorize the continuous response region by dividing it into 3 or 4 bins. Our interest is to determine the sample size needed for the multi-class classification, and use this sample size as the one needed in prediction. This process is similar to the multi-class cases. The continuous variable, *anxiety*, is categorized into three regions according to its distribution with cut-offs at 0.1 and 0.6, while we categorize the *obesity BMI* trait into four regions with cut-offs at the three quantiles of the trait. Figure 3.2 shows plots of sample size required under different thresholds for the two traits. From these figures, the sample size can be estimated. For example, if we set $\widehat{VUS(\infty)} - \widehat{VUS(n)} < 0.05$, then about $3,300$ samples are required for each region of the *anxiety* trait with a total sample size of $10,000$, while when the *obesity BMI* trait is categorized in four regions, about $6,000$ samples are required for each region with a total sample of $24,000$.

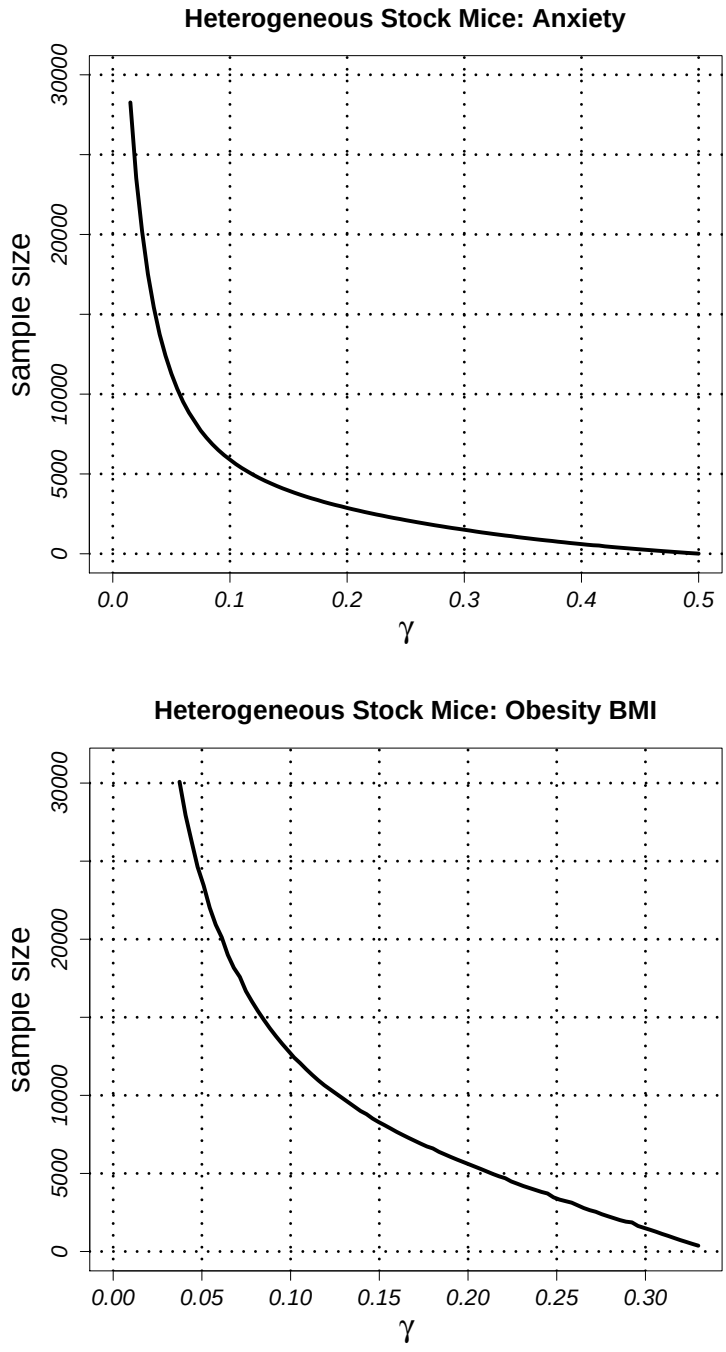

Figure 3.2: The sample size required for the *Stock Mice* Data under different thresholds, γ. Here, $\rho = 1$, $\alpha = 0.01$, $m = 348$. The upper panel is for *anxiety*, which is categorized into three regions according its distribution, with the cut-offs at 0.1 and 0.6, and $\widehat{VUS(\infty)} = 0.6934$. The bottom panel is for *obesity BMI*, which is categorized into four regions using its three quantiles, and $\widehat{VUS(\infty)} = 0.390142$.

3.4 Discussion

We have built an optimal Bayes classifier and a linear classifier based on coded SNP data from two or more classes. For these classifiers, we have considered the two commonly used scalar performance measures, the Area Under the *ROC* curve (AUC) and the Volume Under the *ROC* hyper-Surface (VUS), which allow classifiers to be compared independent of discrimination values. We have illustrated the performance of a sample size determination algorithm which selects the smallest sample size n in each class such that the difference between the (approximate) VUSs for the optimal and the linear classifier, $\widehat{VUS(\infty)} - \widehat{VUS(n)}$, is below a threshold γ. While we have obtained the necessary approximations to the VUS (or AUC) using theory, the simulations and two real data analyses presented here illustrate the practical value of our sample size determination method.

Our method would be useful to scientists who have already collected the SNP data on a set of subjects $n_1, \ldots, n_D$ from each class and is interested in using our classification rules, but is not certain whether the samples at hand are adequate to ensure that $\widehat{VUS(\infty)} - \widehat{VUS(n)} < \gamma$, for a certain value of γ. Given a (user-specified) γ value, our algorithm can find a common $n = n^*$ (say) for each class that satisfies the VUS criterion. If n^* is larger than n_k, for some $k = k_0$, say, then the scientist would have to collect $n^* - n_{k_0}$ samples for the class C_{k_0}; otherwise, the sample at hand can be deemed adequate.

3.5 Appendices

In the following sections, the equation numbers without the prefix, A, correspond to those in the main article.

Appendix 1: Approximation for $PCC(\infty)$

For each fixed $k = 1, 2, ..., D$, in order to prove the assertion, $\vec{\mathbf{Y}}_{l,k} \approx N(\vec{\boldsymbol{\mu}}_{l,k}, \boldsymbol{\Sigma}_{l,k})$ for large l (see equation (3.5) of Section 3.2.2 of the article), we will consider any linear combination

$\vec{\boldsymbol{\beta}}\,'\vec{\mathbf{Y}}_{l,k}$ $(=\sum_{i=1,i\neq k}^{D}\beta_i Y_{l,k}(i))$ and show that

$$\frac{\vec{\boldsymbol{\beta}}\,'\vec{\mathbf{Y}}_{l,k}-\vec{\boldsymbol{\beta}}\,'\vec{\boldsymbol{\mu}}_{l,k}}{\sqrt{\vec{\boldsymbol{\beta}}\,'\boldsymbol{\Sigma}_{l,k}\vec{\boldsymbol{\beta}}}}\Rightarrow N(0,1)\quad \text{as}\ \ l\to\infty. \tag{A.1}$$

To this end, write

$$\vec{\boldsymbol{\beta}}\,'\vec{\mathbf{Y}}_{l,k}=\sum_{i=1,i\neq k}^{D}\beta_i Y_{l,k}(i)=\sum_{i=1,i\neq k}^{D}\sum_{j=1}^{l}\beta_i x_j b_{k,i}^{j}=\sum_{j=1}^{l}(\sum_{i=1,i\neq k}^{D}\beta_i b_{k,i}^{j})x_j.$$

For a fixed $k=1,\ldots,D$, set $Z_j=(\sum_{i=1,i\neq k}^{D}\beta_i b_{k,i}^{j})x_j$ and note that $|Z_j|\leq M$ for some $M>0$, because $|x_j|\leq 2$ and $|b_{k,i}^{j}|\leq log(99)$ since $\theta_{k,j}$ and $\theta_{i,j}\in(0.01,0.5)$. Therefore, by the assumptions in Section 3.2.1, $\{Z_j\}$ is a sequence of independent and bounded random variables with $E(Z_j)=(\sum_{i=1,i\neq k}^{D}\beta_i b_{k,i}^{j})2\theta_{k,j}$ and $Var(Z_j)=(\sum_{i=1,i\neq k}^{D}\beta_i b_{k,i}^{j})^2 2\theta_{k,j}(1-\theta_{k,j})$, when $\vec{x}\in C_k$.

Now, since $\theta_{k,j}(1-\theta_{k,j})>0.01\times 0.5$, by assumption (3) in Section 3.2.1 we have

$$\sum_{j=1}^{l}Var(Z_j)=2\sum_{j=1}^{l}(\sum_{i=1,i\neq k}^{D}\beta_i b_{k,i}^{j})^2\theta_{k,j}(1-\theta_{k,j})>(0.01)\sum_{j=1}^{l}(\sum_{i=1,i\neq k}^{D}\beta_i b_{k,i}^{j})^2\to\infty,$$

as $l\to\infty$. Therefore, the desired result in (A.1) follows from Example 27-4 (also see Problem 27-4) of Billingsley (1995).

Appendix 2: Wald test and its power function

We are interested in deriving a Wald test for testing $H_{0,j}^{k,k'}:\theta_{k,j}=\theta_{k',j}$ versus $H_{1,j}^{k,k'}:\theta_{k,j}\neq\theta_{k',j}$ for each j. For notational convenience, we let $k=1$ and $k'=2$, and $\theta_{1,j}=\theta_1$ and $\theta_{2,j}=\theta_2$ for the derivations below. For each X_j satisfying assumption (2) in Section 3.2.1, let $n_{1k}=\sum_{j=1}^{n_k}I_{\{x_j=0\}}$, $n_{2k}=\sum_{j=1}^{n_k}I_{\{x_j=1\}}$ and $n_{3k}=\sum_{j=1}^{n_k}I_{\{x_j=2\}}$ with $\sum_{i=1}^{3}n_{ik}=n_k$ for $k=1,2$. Then, by assumption 2 in Section 3.2.1 and the independence of the two classes, the likelihood function for a sample of size n_k from each class is:

$$L(\theta_1,\theta_2)=\prod_{k=1}^{2}[(1-\theta_k)^2]^{n_{1k}}[2\theta_k(1-\theta_k)]^{n_{2k}}[\theta_k^2]^{n_{3k}}.$$

Maximizing the log-likelihood, $\log L(\theta_1, \theta_2)$, with respect to (θ_1, θ_2), it can be shown that the maximum likelihood estimator (MLE) of θ_1 and θ_2 are, respectively:

$$\hat{\theta}_1 = \frac{n_{21} + 2n_{31}}{2n_1} \quad \text{and} \quad \hat{\theta}_2 = \frac{n_{22} + 2n_{32}}{2n_2}. \tag{A.2}$$

Also, the Fisher information matrix at (θ_1, θ_2) for $n_k = 1$ is $I(\theta_1, \theta_2) = \begin{pmatrix} \frac{2}{\theta_1(1-\theta_1)} & 0 \\ 0 & \frac{2}{\theta_2(1-\theta_2)} \end{pmatrix}$. Let $2n = n_1 + n_2$. Then, by the asymptotic normality of the MLE, it follows that $\sqrt{n}(\hat{\theta}_1 - \theta_1, \hat{\theta}_2 - \theta_2)' \xrightarrow{d} N_2(\mathbf{0}, I^{-1}(\theta_1, \theta_2))$. Now, since $g(\theta_1, \theta_2) = \theta_1 - \theta_2$ is differentiable at (θ_1, θ_2), it follows from the delta method that $\sqrt{n}[g(\hat{\theta}_1, \hat{\theta}_2) - g(\theta_1, \theta_2)] \xrightarrow{d} N(0, \frac{\theta_1(1-\theta_1)+\theta_2(1-\theta_2)}{2})$. Therefore, under $H_0 : \theta_1 = \theta_2$, the Wald test statistic

$$Q_2 = \frac{2n(\hat{\theta}_1 - \hat{\theta}_2)^2}{\hat{\theta}_1(1 - \hat{\theta}_1) + \hat{\theta}_2(1 - \hat{\theta}_2)} \xrightarrow{d} \chi_1^2 \quad \text{as} \quad n \to \infty, \tag{A.3}$$

where χ_1^2 has chi-square distribution with 1 degree of freedom. However, under $H_a : \theta_1 \neq \theta_2$, say $\theta_1 - \theta_2 = h$, it follows from the above arguments that $Q_2 \xrightarrow{d} \chi_1^2(\lambda^2)$, where $\chi_1^2(\lambda^2)$ has non-central chi-square distribution with the non-centrality parameter, $\lambda^2 = 2nh^2/[\theta_1(1 - \theta_1) + (\theta_1 - h)(1 - \theta_1 + h)]$. Therefore, the power of the Wald test (when $\theta_1 - \theta_2 = h \neq 0$) is:

$$1 - \beta(n_1, n_2, h) \approx P\left(\chi_1^2(\lambda^2) > \chi_{1,(1-\alpha)}^2\right),$$

where $\chi_{1,(1-\alpha)}^2$ is the $(1 - \alpha)$ percentile of χ_1^2. For ease of presentation, we had suppressed the subscript j. For each $j = 1, \ldots, m$, the power of the Wald test for $H_0 : \theta_{k,j} = \theta_{k',j}$ versus $H_1 : \theta_{k,j} \neq \theta_{k',j}$ at $\theta_{k,j} = \theta_{k',j} + h_j$ is denoted by $1 - \beta_j^{k,k'}(n_k, n_{k'}, h_j)$.

Appendix 3: Mean vector and covariance matrix in Section 3.2.3

$$\begin{aligned} E(\hat{b}_{k,k'}^j w_{j,n}(k,k')x_j) &= E\{E(\hat{b}_{k,k'}^j w_{j,n}(k,k')x_j)|w_{j,n}(k,k')\} \\ &\approx E(2\theta_{k,j}\hat{b}_{k,k'}^j w_{j,n}(k,k')) \\ &\approx 2\theta_{k,j} b_{k,k'}^j \tilde{\eta}_j^{k,k'}. \end{aligned}$$

$$
\begin{aligned}
&Cov(w_{j,n}(k,k')\hat{b}^j_{k,k'}x_j, w_{j,n}(k,k'')\hat{b}^j_{k,k'}x_j) \\
&= E(x_j^2 w_{j,n}(k,k')w_{j,n}(k,k'')\hat{b}^j_{k,k'}\hat{b}^j_{k,k''}) - E(w_{j,n}(k,k')\hat{b}^j_{k,k'}x_j)E(w_{j,n}(k,k'')\hat{b}^j_{k,k''}x_j) \\
&= E\{E[x_j^2\hat{b}^j_{k,k'}\hat{b}^j_{k,k''}w_{j,n}(k,k')w_{j,n}(k,k'')|w_{j,n}(k,k')w_{j,n}(k,k'')]\} \\
&\quad - E(w_{j,n}(k,k')\hat{b}^j_{k,k'}x_j)E(w_{j,n}(k,k'')\hat{b}^j_{k,k''}x_j) \\
&= E\{w_{j,n}(k,k')\hat{b}^j_{k,k'}\hat{b}^j_{k,k'}w_{j,n}(k,k'')[2\theta_{k,j}(1-\theta_{k,j})+4\theta^2_{k,j}]\} \\
&\quad - E(w_{j,n}(k,k')\hat{b}^j_{k,k'}x_j)E(w_{j,n}(k,k'')\hat{b}^j_{k,k''}x_j) \\
&\approx \begin{cases}
(b^j_{k,k'})^2[2\theta_{k,j}(1-\theta_{k,j})\tilde{\eta}^{k,k'}_j + 4\theta^2_{k,j}\tilde{\eta}^{k,k'}_j(1-\tilde{\eta}^{k,k'}_j)], \text{if } k'=k'' \\
[2\theta_{k,j}(1-\theta_{k,j})+4\theta^2_{k,j}]b^j_{k,k'}b^j_{k,k''}E(w_{j,n}(k,k')w_{j,n}(k,k'')) \\
\qquad -(2\theta_{k,j}b^j_{k,k'}\tilde{\eta}^{k,k'}_j)(2\theta_{k,j}b^j_{k,k''}\tilde{\eta}^{k,k''}_j), \text{ if } k'\neq k''.
\end{cases}
\end{aligned} \tag{A.4}
$$

Now, we give an approximation for $E(w_{j,n}(k,k')w_{j,n}(k,k''))$ in (A.4). Note first that it can be proved that $\sqrt{2n}(\hat{\theta}_{i,j} - \theta_{i,j}) \Rightarrow N(0, \theta_{i,j}(1-\theta_{i,j}))$ where $i = k, k', k''$ and $\sqrt{2n}(\hat{\theta}_{k,j} - \hat{\theta}_{s,j} - \theta_{k,j} + \theta_{s,j}) \Rightarrow N(0, \theta_{k,j}(1-\theta_{k,j}) + \theta_{s,j}(1-\theta_{s,j}))$, where $s = k', k''$. Now define $T_1 \triangleq \frac{\sqrt{2n}(\hat{\theta}_{k,j}-\hat{\theta}_{k',j})}{\sqrt{\theta_{k,j}(1-\theta_{k,j})+\theta_{k',j}(1-\theta_{k',j})}}$ and $T_2 \triangleq \frac{\sqrt{2n}(\hat{\theta}_{k,j}-\hat{\theta}_{k'',j})}{\sqrt{\theta_{k,j}(1-\theta_{k,j})+\theta_{k'',j}(1-\theta_{k'',j})}}$, then from (A.3) we have $Q_2(\theta_{k,j}, \theta_{k',j}) \triangleq T_1^2$ and $Q_2(\theta_{k,j}, \theta_{k'',j}) \triangleq T_2^2$ and recall that $H^{k,k'}_{0,j} : \theta_{k,j} = \theta_{k',j}$ and $H^{k,k''}_{0,j} : \theta_{k,j} = \theta_{k'',j}$. Then, we have

$$
\begin{aligned}
&E(w_{j,n}(k,k')w_{j,n}(k,k'')) \\
&= P(reject\ H^{k,k'}_{0,j} \cap reject\ H^{k,k''}_{0,j}) \\
&= P(\{Q_2(\theta_{k,j},\theta_{k',j}) > \chi^2_{1-\alpha}(1)\} \cap \{Q_2(\theta_{k,j},\theta_{k'',j}) > \chi^2_{1-\alpha}(1)\}) \\
&= P(|T_1| > \sqrt{\chi^2_{1-\alpha}(1)} \cap |T_2| > \sqrt{\chi^2_{1-\alpha}(1)})
\end{aligned}
$$

For k, k', k'', we can mimic the arguments leading to (A.2) and (A.3), and show that (T_1, T_2) is asymptotically multivariate normal. Note that the means, variances, and covariances of T_1 and T_2 are given by:

$$
E(T_1) = \frac{\sqrt{2n}(\theta_{k,j} - \theta_{k',j})}{\sqrt{\theta_{k,j}(1-\theta_{k,j}) + \theta_{k',j}(1-\theta_{k',j})}}
$$

$$E(T_2) = \frac{\sqrt{2n}(\theta_{k,j} - \theta_{k'',j})}{\sqrt{\theta_{k,j}(1-\theta_{k,j}) + \theta_{k'',j}(1-\theta_{k'',j})}}$$

$$Var(T_1) = Var(T_2) = 1$$

$$Cov(T_1, T_2) = \frac{\theta_{k,j}(1-\theta_{k,j})}{\sqrt{[\theta_{k,j}(1-\theta_{k,j}) + \theta_{k',j}(1-\theta_{k',j})][\theta_{k,j}(1-\theta_{k,j}) + \theta_{k'',j}(1-\theta_{k'',j})]}}.$$

Appendix 4: Example: sample size determination for three classes

Here, we assume that $D = 3$ and obtain an expression for $PCC(\infty), PCC(n)$ and $VUS(\infty)$.

Calculation of PCC(∞):

$$PCC(\infty) = \pi_1 P(\sum_{j=1}^{l} x_j b_{1,2}^j > K_{1,2},\ \sum_{j=1}^{l} x_j b_{1,3}^j > K_{1,3}) + \pi_2 P(\sum_{j=1}^{l} x_j b_{2,1}^j > K_{2,1},\ \sum_{j=1}^{l} x_j b_{2,3}^j > K_{2,3})$$
$$+ \pi_3 P(\sum_{j=1}^{l} x_j b_{3,1}^j > K_{3,1},\ \sum_{j=1}^{l} x_j b_{3,2}^j > K_{3,2}).$$

Let

$$K_1 \triangleq K_{1,2} = -K_{2,1}$$

$$K_2 \triangleq K_{2,3} = -K_{3,2}$$

$$K_{1,3} = K_1 + K_2$$

$$K_{3,1} = -(K_1 + K_2).$$

Then, rewrite $PCC(\infty)$ as

$$PCC(\infty) = \pi_1 P(\sum_{j=1}^{l} x_j b_{1,2}^j > K_1,\ \sum_{j=1}^{l} x_j b_{1,3}^j > (K_1 + K_2)) + \pi_2 P(\sum_{j=1}^{l} x_j b_{2,1}^j > -K_1,\ \sum_{j=1}^{l} x_j b_{2,3}^j > K_2)$$
$$+ \pi_3 P(\sum_{j=1}^{l} x_j b_{3,1}^j > -(K_1 + K_2),\ \sum_{j=1}^{l} x_j b_{3,2}^j > -K_2)$$
$$\approx \pi_1 \tilde{\Phi}((K_1, K_1 + K_2)'; \vec{\boldsymbol{\mu}}_{l,1}, \boldsymbol{\Sigma}_{l,1}) + \pi_2 \tilde{\Phi}((-K_1, K_2)'; \vec{\boldsymbol{\mu}}_{l,2}, \boldsymbol{\Sigma}_{l,2})$$
$$+ \pi_3 \tilde{\Phi}((-(K_1 + K_2), -K_2)'; \vec{\boldsymbol{\mu}}_{l,3}, \boldsymbol{\Sigma}_{l,3}), \qquad (A.5)$$

where

$$\vec{\boldsymbol{\mu}}_{l,1} = \begin{pmatrix} \sum_{j=1}^{l} 2\theta_{1,j} b_{1,2}^{j} \\ \sum_{j=1}^{l} 2\theta_{1,j} b_{1,3}^{j} \end{pmatrix}$$

$$\vec{\boldsymbol{\mu}}_{l,2} = \begin{pmatrix} \sum_{j=1}^{l} 2\theta_{2,j} b_{2,1}^{j} \\ \sum_{j=1}^{l} 2\theta_{2,j} b_{2,3}^{j} \end{pmatrix}$$

$$\vec{\boldsymbol{\mu}}_{l,3} = \begin{pmatrix} \sum_{j=1}^{l} 2\theta_{3,j} b_{3,1}^{j} \\ \sum_{j=1}^{l} 2\theta_{3,j} b_{3,2}^{j} \end{pmatrix}$$

$$\boldsymbol{\Sigma}_{l,1} \triangleq \begin{pmatrix} \sum_{j=1}^{l} 2(b_{1,2}^{j})^2 \theta_{1,j}(1-\theta_{1,j}) & \sum_{j=1}^{l} 2b_{1,2}^{j} b_{1,3}^{j} \theta_{1,j}(1-\theta_{1,j}) \\ \sum_{j=1}^{l} 2b_{1,3}^{j} b_{1,2}^{j} \theta_{1,j}(1-\theta_{1,j}) & \sum_{j=1}^{l} 2(b_{1,3}^{j})^2 \theta_{1,j}(1-\theta_{1,j}) \end{pmatrix}$$

$$\boldsymbol{\Sigma}_{l,2} \triangleq \begin{pmatrix} \sum_{j=1}^{l} 2(b_{2,1}^{j})^2 \theta_{2,j}(1-\theta_{2,j}) & \sum_{j=1}^{l} 2b_{2,1}^{j} b_{2,3}^{j} \theta_{2,j}(1-\theta_{2,j}) \\ \sum_{j=1}^{l} 2b_{2,3}^{j} b_{2,1}^{j} \theta_{2,j}(1-\theta_{2,j}) & \sum_{j=1}^{l} 2(b_{2,3}^{j})^2 \theta_{2,j}(1-\theta_{2,j}) \end{pmatrix}$$

$$\boldsymbol{\Sigma}_{l,3} \triangleq \begin{pmatrix} \sum_{j=1}^{l} 2(b_{3,1}^{j})^2 \theta_{3,j}(1-\theta_{3,j}) & \sum_{j=1}^{l} 2b_{3,1}^{j} b_{3,2}^{j} \theta_{3,j}(1-\theta_{3,j}) \\ \sum_{j=1}^{l} 2b_{3,1}^{j} b_{3,2}^{j} \theta_{3,j}(1-\theta_{3,j}) & \sum_{j=1}^{l} 2(b_{3,2}^{j})^2 \theta_{3,j}(1-\theta_{3,j}) \end{pmatrix}.$$

Calculation of PCC(n):

Note from (3.4) in Section 3.2.2 that

$$\vec{X} \in C_1 \; if \; \{\sum_{j=1}^{m} \hat{b}_{1,2}^{j} w_{j,n}(1,2) x_j > \tilde{K_1}\} \; \& \; \{\sum_{j=1}^{m} \hat{b}_{1,3}^{j} w_{j,n}(1,3) x_j > \tilde{K_1} + \tilde{K_2}\},$$

$$\vec{X} \in C_2 \; if \; \{\sum_{j=1}^{m} \hat{b}_{2,1}^{j} w_{j,n}(2,1) x_j > -\tilde{K_1}\} \; \& \; \{\sum_{j=1}^{m} \hat{b}_{2,3}^{j} w_{j,n}(2,3) x_j > \tilde{K_2}\},$$

$$\vec{X} \in C_3 \; if \; \{\sum_{j=1}^{m} \hat{b}_{3,1}^{j} w_{j,n}(3,1) x_j > -(\tilde{K_1} + \tilde{K_2})\} \; \& \; \{\sum_{j=1}^{m} \hat{b}_{3,2}^{j} w_{j,n}(3,2) x_j > -\tilde{K_2}\},$$

and

$$\begin{aligned}
PCC(n) &= \pi_1 P(\sum_{j=1}^{m} \hat{b}_{1,2}^{j} w_{j,n}(1,2) x_j > \tilde{K_1}, \; \sum_{j=1}^{m} \hat{b}_{1,3}^{j} w_{j,n}(1,3) x_j > \tilde{K_1} + \tilde{K_2}) \\
&\quad + \pi_2 P(\sum_{j=1}^{m} \hat{b}_{2,1}^{j} w_{j,n}(2,1) x_j > -\tilde{K_1}, \; \sum_{j=1}^{m} \hat{b}_{2,3}^{j} w_{j,n}(2,3) x_j > \tilde{K_2}) \\
&\quad + \pi_3 P(\sum_{j=1}^{m} \hat{b}_{3,1}^{j} w_{j,n}(3,1) x_j > -(\tilde{K_1} + \tilde{K_2}), \; \sum_{j=1}^{m} \hat{b}_{3,2}^{j} w_{j,n}(3,2) x_j > -\tilde{K_2}) \\
&\approx \pi_1 \tilde{\Phi}((\tilde{K_1}, \tilde{K_1} + \tilde{K_2})'; \tilde{\vec{\boldsymbol{\mu}}}_{l,1}, \tilde{\boldsymbol{\Sigma}}_{l,1}) + \pi_2 \tilde{\Phi}((-\tilde{K_1}, \tilde{K_2})'; \tilde{\vec{\boldsymbol{\mu}}}_{l,2}, \tilde{\boldsymbol{\Sigma}}_{l,2}) \\
&\quad + \pi_3 \tilde{\Phi}((-(\tilde{K_1} + \tilde{K_2}), -\tilde{K_2})'; \tilde{\vec{\boldsymbol{\mu}}}_{l,3}, \tilde{\boldsymbol{\Sigma}}_{l,3}),
\end{aligned}$$

where

$$\tilde{\vec{\boldsymbol{\mu}}}_{l,1} = \begin{pmatrix} 2\sum_{j=1}^{m} \theta_{1,j} \tilde{\eta}_j^{1,2} \\ 2\sum_{j=1}^{m} \theta_{1,j} \tilde{\eta}_j^{1,3} \end{pmatrix}$$

$$\tilde{\vec{\boldsymbol{\mu}}}_{l,2} = \begin{pmatrix} 2\sum_{j=1}^{m} \theta_{2,j} \tilde{\eta}_j^{2,1} \\ 2\sum_{j=1}^{m} \theta_{2,j} \tilde{\eta}_j^{2,3} \end{pmatrix}$$

$$\tilde{\vec{\boldsymbol{\mu}}}_{l,3} = \begin{pmatrix} 2\sum_{j=1}^{m} \theta_{3,j} \tilde{\eta}_j^{3,1} \\ 2\sum_{j=1}^{m} \theta_{3,j} \tilde{\eta}_j^{3,2} \end{pmatrix},$$

and the 2×2 variance-covariance matrices (written in a vector form due to the length of each expression) are given by:

$$\tilde{\boldsymbol{\Sigma}}_{l,1} = \begin{pmatrix} \sum_{j=1}^{m} (b_{1,2}^{j})^2 [2\theta_{1,j}(1-\theta_{1,j})\tilde{\eta}_j^{1,2} + 4\theta_{1,2}^2 \tilde{\eta}_j^{1,2}(1-\tilde{\eta}_j^{1,2})] \\ \sum_{j=1}^{m} [2\theta_{1,j}(1-\theta_{1,j}) + 4\theta_{1,j}^2] b_{1,2}^{j} b_{1,3}^{j} E(w_{j,n}(1,2)w_{j,n}(1,3)) - \sum_{j=1}^{m} 4\theta_{1,j}^2 b_{1,2}^{j} b_{1,3}^{j} \tilde{\eta}_j^{1,2} \tilde{\eta}_j^{1,3}] \\ \sum_{j=1}^{m} [2\theta_{1,j}(1-\theta_{1,j}) + 4\theta_{1,j}^2] b_{1,2}^{j} b_{1,3}^{j} E(w_{j,n}(1,2)w_{j,n}(1,3)) - \sum_{j=1}^{m} 4\theta_{1,j}^2 b_{1,2}^{j} b_{1,3}^{j} \tilde{\eta}_j^{1,2} \tilde{\eta}_j^{1,3}] \\ \sum_{j=1}^{m} (b_{1,3}^{j})^2 [2\theta_{1,j}(1-\theta_{1,j})\tilde{\eta}_j^{1,3} + 4\theta_{1,3}^2 \tilde{\eta}_j^{1,3}(1-\tilde{\eta}_j^{1,3})] \end{pmatrix}$$

$$\tilde{\boldsymbol{\Sigma}}_{l,2} = \begin{pmatrix} \sum_{j=1}^{m} (b_{2,1}^{j})^2 [2\theta_{2,j}(1-\theta_{2,j})\tilde{\eta}_j^{2,1} + 4\theta_{2,1}^2 \tilde{\eta}_j^{2,1}(1-\tilde{\eta}_j^{2,1})] \\ \sum_{j=1}^{m} [2\theta_{2,j}(1-\theta_{2,j}) + 4\theta_{2,j}^2] b_{2,1}^{j} b_{2,3}^{j} E(w_{j,n}(2,1)w_{j,n}(2,3)) - \sum_{j=1}^{m} 4\theta_{2,j}^2 b_{2,1}^{j} b_{2,3}^{j} \tilde{\eta}_j^{2,1} \tilde{\eta}_j^{2,3}] \\ \sum_{j=1}^{m} [2\theta_{2,j}(1-\theta_{2,j}) + 4\theta_{2,j}^2] b_{2,2}^{j} b_{2,3}^{j} E(w_{j,n}(2,1)w_{j,n}(2,3)) - \sum_{j=1}^{m} 4\theta_{2,j}^2 b_{2,1}^{j} b_{2,3}^{j} \tilde{\eta}_j^{2,1} \tilde{\eta}_j^{2,3}] \\ \sum_{j=1}^{m} (b_{2,3}^{j})^2 [2\theta_{2,j}(1-\theta_{2,j})\tilde{\eta}_j^{2,3} + 4\theta_{2,3}^2 \tilde{\eta}_j^{2,3}(1-\tilde{\eta}_j^{2,3})] \end{pmatrix}$$

$$\tilde{\boldsymbol{\Sigma}}_{l,3} = \begin{pmatrix} \sum_{j=1}^{m} (b_{3,1}^{j})^2 [2\theta_{3,j}(1-\theta_{3,j})\tilde{\eta}_j^{3,2} + 4\theta_{3,j}^2 \tilde{\eta}_j^{3,2}(1-\tilde{\eta}_j^{3,2})] \\ \sum_{j=1}^{m} [2\theta_{3,j}(1-\theta_{3,j}) + 4\theta_{3,j}^2] b_{3,1}^{j} b_{3,2}^{j} E(w_{j,n}(3,1)w_{j,n}(3,2)) - \sum_{j=1}^{m} 4\theta_{3,j}^2 b_{3,1}^{j} b_{3,2}^{j} \tilde{\eta}_j^{3,1} \tilde{\eta}_j^{3,2}] \\ \sum_{j=1}^{m} [2\theta_{3,j}(1-\theta_{3,j}) + 4\theta_{3,j}^2] b_{3,1}^{j} b_{3,2}^{j} E(w_{j,n}(3,1)w_{j,n}(3,2)) - \sum_{j=1}^{m} 4\theta_{3,j}^2 b_{3,1}^{j} b_{3,2}^{j} \tilde{\eta}_j^{3,1} \tilde{\eta}_j^{3,2}] \\ \sum_{j=1}^{m} (b_{3,2}^{j})^2 [2\theta_{3,j}(1-\theta_{3,j})\tilde{\eta}_j^{3,1} + 4\theta_{3,j}^2 \tilde{\eta}_j^{3,2}(1-\tilde{\eta}_j^{3,2})] \end{pmatrix}.$$

Calculation of $VUS(\infty)$ and $VUS(n)$:

If we denote $N_2(x_1, x_2; \vec{\boldsymbol{\mu}}, \boldsymbol{\Sigma})$ as the two-dimensional normal density function with mean $\vec{\boldsymbol{\mu}}$ and variance-covariance matrix $\boldsymbol{\Sigma}$, then from (A.5) and (3.13) of Section 3.2.4, the right side of $PCC(\infty)$ involves

$$\begin{aligned} \xi_{1,1} &= \int_{K_1}^{\infty} \int_{K_1+K_2}^{\infty} N_2(x_1, x_2; \vec{\boldsymbol{\mu}}_{l,1}, \boldsymbol{\Sigma}_{l,1}) dx_1 dx_2 \\ \xi_{2,2} &= \int_{-K_1}^{\infty} \int_{K_2}^{\infty} N_2(x_1, x_2; \vec{\boldsymbol{\mu}}_{l,2}, \boldsymbol{\Sigma}_{l,2}) dx_1 dx_2 \\ \xi_{3,3} &= \int_{-K_1-K_2}^{\infty} \int_{-K_2}^{\infty} N_2(x_1, x_2; \vec{\boldsymbol{\mu}}_{l,3}, \boldsymbol{\Sigma}_{l,3}) dx_1 dx_2. \end{aligned}$$

Then, from (3.13) of Section 3.2.4 we have

$$VUS(\infty) = \int_0^1 \int_0^1 \xi_{1,1}(K_1, K_2) d\xi_{2,2}(K_1, K_2) d\xi_{3,3}(K_1, K_2).$$

Appendix 5: Monte Carlo Simulations using AUC

To compare the performance of our linear classifier with another classifier in the literature, such as the SVM, we also computed the $AUC(n)$ values corresponding to the SVM for the same simulation setup as the one described in Table 3.3 of Liu et al. (2012). For ROC and AUC calculations, we consider the special case, $\vec{\theta_1} = (\theta_1, \ldots, \theta_1)'$ and $\vec{\theta_2} = (\theta_2, \ldots, \theta_2)'$ with $\theta_1 > \theta_2$. These are given in Figure 3.3 below. Note that, unlike our linear classifier, there is no approximate formula available to calculate the $AUC(n)$ for SVM. Therefore, we cannot compare $AUC(n)$ values for our linear classifier (or the $AUC(\infty)$ values) with $AUC(n)$ values for SVM. Figure 3.3 shows that the $ROC_$MC values are essentially same as those for the ROC for the SVM_MC. This says that our linear classifier is as good as or slightly better than the SVM. Table 3.3 compares the approximate values of $AUC(n)$, denoted by $\widehat{AUC(n)}$, with the Monte Carlo based estimates, $\widehat{AUC(n)}$MC, for various specifications. To obtain $\widehat{AUC(n)}$MC values, for each specification in Table 3.3, we simulated a *training* data and a *testing* data of SNPs, each having the same sample sizes. The training data was used to build the linear classifier, while the testing data was used to determine the frequency of correct classification of the linear classifier. This process was repeated 200 times in order to compute the average correct classification frequency, $\widehat{AUC(n)}$MC, given in Table 3.3. It is evident from Table 3.3 that the $\widehat{Bias} = \widehat{AUC(n)}\text{MC} - \widehat{AUC(n)}$ is negligible in most cases, thereby validating the use of our approximation for $AUC(n)$. Also note that both $\widehat{AUC(n)}$MC and $\widehat{AUC(n)}$ are close to $\widehat{AUC(\infty)}$, approximate values of $AUC(\infty)$.

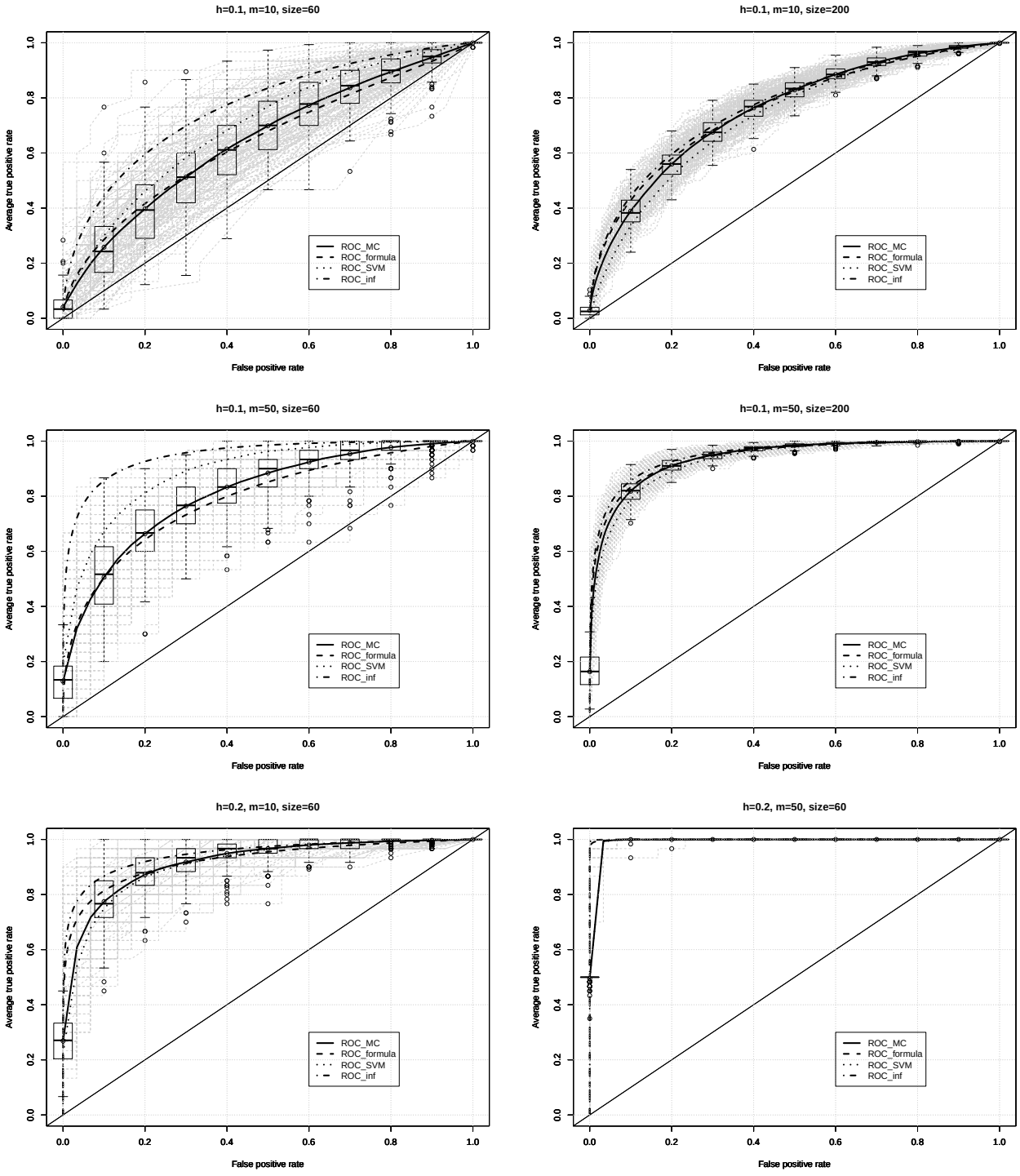

Figure 3.3: ROC curves for optimal classification, linear classification, Monte Carlo simulation and SVM. The shade is the ROC curve for each simulation. $\alpha = 0.1$, $\rho = 1$

Table 3.3: Performance of Optimal and Linear classifiers. The values of $\widehat{AUC(n)}$ and $\widehat{AUC(n)}$MC are close to each other for various model specifications. Here, $\theta_1 = 0.3, h = \theta_1 - \theta_2$, $Size = 2n$ (n for C_1, n for C_2), m is the number of independent SNPs, $\alpha = 0.01$ is the significant level for Wald tests in Section 3.2.3, and $\rho = 1$ is the percentage of the significant SNPs.

h	m	Size	$\widehat{AUC(\infty)}$	$\widehat{AUC(n)}$	$\widehat{AUC(n)}$MC	$\widehat{Bias}$
0.01	10	60	0.5276	0.5021	0.5032	0.0011
0.01	10	200	0.5276	0.5022	0.5022	0
0.01	10	400	0.5276	0.5024	0.5016	-0.0008
0.01	50	60	0.5616	0.5047	0.5183	0.0136
0.01	50	200	0.5616	0.505	0.5111	0.0061
0.01	50	400	0.5616	0.5054	0.5079	0.0025
0.01	200	60	0.6218	0.5094	0.5373	0.0279
0.01	200	200	0.6218	0.5099	0.5217	0.0118
0.01	200	400	0.6218	0.5107	0.5169	0.0062
0.05	10	60	0.6386	0.5171	0.5122	-0.0049
0.05	10	200	0.6386	0.5292	0.5288	-0.0004
0.05	10	400	0.6386	0.5442	0.5439	-0.0003
0.05	50	60	0.7861	0.5382	0.5426	0.0044
0.05	50	200	0.7861	0.565	0.5752	0.0102
0.05	50	400	0.7861	0.5979	0.6197	0.0218
0.05	200	60	0.9436	0.5761	0.5864	0.0103
0.05	200	200	0.9436	0.6283	0.6585	0.0302
0.05	200	400	0.9436	0.6901	0.7367	0.0466
0.2	10	60	0.9488	0.8594	0.8746	0.0152
0.2	10	200	0.9488	0.9471	0.9452	-0.0019
0.2	10	400	0.9488	0.9488	0.9464	-0.0024
0.2	50	60	0.9999	0.992	0.9944	0.0024
0.2	50	200	0.9999	0.9999	0.9999	0
0.2	50	400	0.9999	0.9999	0.9999	0
0.2	200	60	1	1	1	0
0.2	200	200	1	1	1	0
0.2	200	400	1	1	1	0

3.6 References

[1] Billingsely, P. (1995). *Probability and Measure*. Wiley, New York.

[2] De Valpine,P., Bitter,H.M., Brown,M.P.S., and Heller,J. (2009) A simulation-approximation approach to sample size planning for high-dimensional classification studies. *Biostatistics*, **10**, 424-435.

[3] de Roos,A.P.W., Hayes,B.J. and Goddard,M.E. (2009) Reliability of Genomic Predictions Across Multiple Populations. *Genetics*, **183**, 1545-1553.

[4] Dobbin,K.K. and Simon,R.M. (2005) Sample size determination in microarray experiments for class comparison and prognostic classification. *Biostatistics*, **6**, 27-38.

[5] Dobbin,K.K. and Simon,R.M. (2007) Sample size planning for developing classifiers using high-dimensional DNA microarray data. *Biostatistics*, **8**, 101-117.

[6] Dobbin,K.K., Zhao,Y., and Simon,R.M. (2008) How large a training set is needed to develop a classifier for microarray data. *Clin Cancer Res*, **14**, 108-114.

[7] Fawcett,T., (2005). An introduction to ROC analysis. *Pattern Recognition, Lett.*, **27**, 861-874.

[8] Guzzetta,G., Jurman,G., and Furlanello,C. (2010). A machine learning pipeline for quantitative phenotype prediction from genotype data. *BMC Bioinformatics*, **11**(Suppl 8), S3.

[9] Lee,S.H., van der Werf,J.H.J., Hayes,B.J., Goddard,M.E., and Visscher,P.M. (2008) Predicting Unobserved Phenotypes for Complex Traits from Whole-Genome SNP Data. *Plos Genet*, **4**, e1000231.

[10] Liu,X., Wang,Y., Rekhaya,R., and Sriram,T.N. (2012) Sample size determination for classifiers based on single-nucleotide polymorphisms. *Biostat.*, **13**, 217-227

[11] Lorenzana,R.E., Bernardo,R. (2009) Accuracy of genotypic value predictions for marker-based selection in biparental plant populations. *Theor. Appl. Genet.*, **120**, 151-161.

[12] Metz,C., (1978). Basic principles of ROC analysis. *Seminars in Nucl. Med.* **3**.

[13] Mukherjee,S., Tamayo,P., Rogers,S., Rifkin,R., Engle,A., Campbell,C., Golub,T.R., and Mesirov,J.P. (2003) Estimating dataset size requirements for classifying DNA microarray data. *J Comput Biol*, **10**, 119-142.

[14] Nunkesser,R., Bernholt,T., Schwender,H., Ickstadt,K., and Wegener,I. (2007) Detecting high-order interactions of single nucleotide polymorphisms using genetic programming. *Bioinformatics*, **23**, 3280-3288.

[15] Landgrebe, T. and Duin, R. P. W. (2007) Approximating the multiclass ROC by pairwise analysis. *Pattern Recognition Letters*, **28**, 1747-1758.

[16] Landgrebe, T. and Paclik, P. (2010) The ROC skeleton for multiclass ROC estimation. *Pattern Recognition Letters*, **31**, 949-958.

[17] Wray,N.R., Goddard,M.E., and Visscher.P.M. (2007) Prediction of individual genetic risk to disease from genome-wide association studies. *Genome Res.*, **17**, 1520-1528.

[18] Wei,Z., Wang,K., Qu,H.Q., Zhang,H., Bradfield,J., et al. (2009) From Disease Association to Risk Assessment: An Optimistic View from Genome-Wide Association Studies on Type 1 Diabetes. *PLoS Genet.*, **5**, e1000678.

[19] Zhou, N. and Wang, L. (2007) Effective selection of informative SNPs and classification on the HapMap genotype data *BMC Bioinformatics*, **8**, 484-492.

Chapter 4

Overall Conclusions

In this dissertation, we observe a coded SNP data vector $\vec{x} = (x_1, \ldots, x_p)'$ of dimension p, (coding is done according to the number of minor alleles) and collect a sample of size $n_1, \ldots, n_D$ from classes $C_1, \ldots, C_D; D \geq 2$, respectively. Here, p is much large than $n_k, k = 1, \ldots, D$. Based on the Hardy-Weinberg equilibrium assumption, we have derived an optimal (Bayes) classifer and a linear classifier. For each of these classifiers, we have provided an asymptotic approximation for the probability of correct classification (PCC), as $p \to \infty$. These approximations are then validated through a variety of Monte Carlo simulations.

When the number of classes is 2, a sample size determination algorithm based on the criterion which ensures that the difference between the two approximate $PCCs$ is below a threshold is given and its effectiveness is also illustrated via simulations. For the HapMap data on Chinese and Japanese populations, a linear classifier is built using 51 independent SNPs, and the required total sample sizes are determined using our algorithm, as the threshold varies.

We have also extended the problem of sample size determination to scenarios where there are two or more classes, that is, $D \geq 2$. As in the two-class scenario, for coded SNP data, we have once again derived an optimal (Bayes) classifier and a linear classifier, based on the Hardy-Weinberg equilibrium assumption. Unlike the two-class scenario, for the multi-class classifiers, we have considered the two commonly used scalar performance measures, the Area Under the ROC curve (AUC) and the Volume Under the ROC hyper-Surface (VUS), which allow classifiers to be compared independent of discrimination values. Once again, we have obtained an approximation to the AUC/VUS corresponding to the optimal and linear

classifier. These approximations are validated through a series of Monte Carlo simulations. Finally, a sample size determination algorithm is developed for the multi-class scenario based on the criterion which ensures that the difference between the two approximate $AUCs/VUSs$ is below a threshold. For the HapMap data with 3 and 4 populations, respectively, a linear classifier is built using 92 independent SNPs, and the required total sample sizes are determined using our algorithm, as the threshold varies.

Furthermore, our sample size determination method is applied to a prediction problem invloving values of traits. Here, the region of trait is categorized into several sub-regions, thereby creating a multi-class classification scenario. We have applied our multi-class sample size determination methodolgy to the Heterogeneous Stock Mice Data, where the traits *Anxiety* and *Obesity BMI*, are categorized into 3 or 4 groups. In this case, a linear classifier is built based on 348 SNPs, and the required sample sizes are also estimated.

In summary, we have developed an asymptotic method to estimate the learning curve of SNP classifiers. It is illustrated that a required sample size can be obtained from the estimated learning curve.