University of Alberta

Library Release Form

Name of Author: Wei Wei

Title of Thesis: Breeding value estimation and quantitative trait loci detection by Machine Learning methods based on high dimensional Single Nucleotide Polymorphisms dataset

Degree: Master of Science

Year this Degree Granted: 2008

Wei Wei
121 Athabasca Hall, University of Alberta
Edmonton, Alberta
Canada, T6G 2E8

Date: ____________

Find a bug in a program, and fix it, and the program will work today. Show the program how to find and fix a bug, and the program will work forever.

– Oliver G. Selfridge.

University of Alberta

Breeding value estimation and quantitative trait loci detection by Machine Learning methods based on high dimensional Single Nucleotide Polymorphisms dataset

by

Wei Wei

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

in

Machine Learning

Department of Computing Science

Edmonton, Alberta
Fall 2008

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Breeding value estimation and quantitative trait loci detection by Machine Learning methods based on high dimensional Single Nucleotide Polymorphisms dataset** submitted by Wei Wei in partial fulfillment of the requirements for the degree of **Master of Science** in *Machine Learning.*

____

Russ Greiner

____

David Wishart

Date: ________________

To Xiaoqian Shi,
You are my everything.

Abstract

A *Quantitative Trait Locus* (QTL) is a region of DNA that is associated with a particular phenotypic trait. *QTL mapping* is the statistical study that relates the alleles that occur in a locus to the associated phenotypes. If we know the QTLs that affect the economically important traits in the breeding industry of dairy cattle, we could greatly improve the estimation of breeding values, which would in turn lead to more accurate selection of diary sires for breeding. With the advances in DNA chip technology and the discovery of thousands of *single nucleotide polymorphisms* (SNPs) in genome-sequencing projects, we can now identify the QTL associated with traits of interest based on the SNP information.

In this study, we consider the challenge of learning the QTL mapping for predicting important traits that are then turned into breeding values using the SNPs dataset. This is especially challenging due to the high dimensionality of the dataset. We examine the use of two machine-learning kernel methods, *Support Vector Machine* (SVM) and *Gaussian Process* (GP), as well as several statistical methods — including *partial least square regression* (PLS) and *LASSO*. We also explore several feature selection techniques to identify the SNPs associated with the QTL affecting the traits for prediction, including correlation-based feature selection, logic regression, M5 prime for linear regression and haplotype blocks.

We focus on a dataset from a diary-industry breeding program, where 1341 SNPs are genotyped of 462 dairy sires to predict 5 economically important traits. Our empirical results indicate that the average correlation between prediction and true value of these 5 traits is about 0.56 using GP, our best predictor. The results also suggest that the performance of the two kernel methods is better than that of the other statistical methods based on correlation and root-mean square error performance criteria. However, the feature selection methods we tried failed to identify the most relevant SNPs of the traits in this dataset.

Acknowledgements

I would like to take this opportunity to extend my deepest gratitude to my supervisor, Professor Russ Greiner, for all the guidance and encouragement he has offered me throughout the period of this research. I am also grateful to Prof. Paul Stothard, Prof. Zhiquan Wang, and Dr. Jason Grant for their thought-provoking advice and support for the research project. Finally, special thanks to my wife for her patience, encouragement and lifelong support.

Table of Contents

List of Tables

List of Figures

List of Symbols

$\boldsymbol{x}$	A vector
x_i	The ith element of vector $\boldsymbol{x}$
$\overline{x}$	The mean of vector $\boldsymbol{x}$
X	A covariate
$\mathbf{X}$	A matrix
$\mathbf{X}^T$	The transpose of matrix $\mathbf{X}$
$\mathbf{X}^{-1}$	The inverse of matrix $\mathbf{X}$
$\boldsymbol{X}_i$	The ith row of matrix $\mathbf{X}$ with p columns $\boldsymbol{X}_i =< X_{1i}, \cdots, X_{pi} >$
x_{ij}	The element of matrix $\mathbf{X}$, located at the ith row and jth column
c	Class label
p	Number of covariates
n	Number of observations
$\boldsymbol{\beta}$	Coefficients vector
$\hat{\boldsymbol{\beta}}$	Predictions of coefficients vector
$p(e)$	The probability of event e
$A \perp B$	Event A is independent of event B
$A \cdot B$	Event A is dependent on event B
$\mathbb{D}$	Training set
$\mathcal{N}(\mu, \sigma^2)$	A normal distribution with μ mean and σ^2 variance
λ	Complexity parameter of regulation method

Chapter 1

Introduction

In 1858, Darwin published his famous *On the Origin of Species*, which introduced one of the cornerstones of modern biology, Natural Selection. Natural selection is the mechanism of evolution, the process in nature by which only the organisms that are best adapted to their environment tend to survive and transmit their genetic characteristics to the next generation. Individuals less well adapted to their environment tend to be eliminated, where the environment represents the combined biological and physical influences.

In fact, Darwin thought of natural selection by analogy to how farmers select crops or livestock for breeding, which he called Artificial Selection, where we, humans, act as the environmental pressure. Artificial selection is of great interest to the breeding industry through its use in the improvement of the desired traits that an animal breeder wishes to develop, like the number of eggs laid by hens, the meat properties of bullocks, or the milk yield of cows.

Some traits can fall into a few distinct phenotypic classes, like eye color, while many traits of biological and economical interest are continuous and are often given a quantitative value. The improvement of these "quantitative traits" has been an important goal for many animal-breeding programs. Although today's "conventional animal breeding methods" have evolved considerably since their emergence in ancient times, they continue to revolve around three basic steps: (1) generation of a population of animals having desirable traits, (2) evaluation and selection of superior individuals, (3) recombination of the superior individuals to generate a new population for subsequent cycles of selection and improvement [1].

This type of methods, however, requires a large amount of labor, land, time and money. Therefore breeders are interested in identifying the most promising individuals as early as possible in the selection process. Recent studies [92,93,94] have shown that quantitative traits are likely to be controlled by a fairly large, but unknown, set of genes. Fortunately however, typically only a few of these genes have a large effect. Such genes are called major genes, and their locations are called the *quantitative trait loci*, or QTL. Furthermore, the process of finding the QTL for a given trait is called *QTL mapping*. (Although the term QTL strictly applies to any genes of an effect, in practice it refers only to major genes, as only these will be large enough to be detected and mapped on the genome.) Because the QTL alleles do not change over the life of an individual, they can be obtained when a potential breeding animal is very young. Hence, if breeders could identify those QTL alleles that contribute to a high value of an important trait, they could greatly accelerate the selection process.

Figure 1.1 illustrates that QTL constitute only some of the many genes that affect phenotype. The other relevant genes are termed polygenes. Variation at the polygenes jointly with polymorphism at the QTL determines total genetic variation. Although QTL effects explain only a part of genetic differences between animals, knowledge of the genes located at QTL could greatly assist in estimating an animal's true genotype. Information available at QTL therefore adds to accuracy of estimation of breeding value.

Figure 1.1 suggests that the value or the allelic forms at individual QTL are known. In practice, this is rarely the case, as the exact gene locations are often unknown. That is, currently there are few examples where QTL effects can be directly determined, but knowledge in this area is rapidly developing [15,16,17,18]. Most of the QTLs known today are known based on genetic markers.

Genetic markers are landmarks at the genome that are chosen for their proximity to QTL. Given the pedigree information, we cannot actually observe inheritance at the QTL itself, but we can observe inheritance at the marker, which is linked to the QTL. Thus the selection is based on the marker linked to the QTL, instead of the QTL itself, with the assumption that the linked allele is associated with QTL of interest. Figure 1.2 shows the principle of inheritance of a marker and a linked QTL. We can identify the marker genotype (M/m) but not the QTL genotype (Q/q). The last is really what we want to know because of its effect on economically important traits. The selection process described above is formally called Marker-Assisted Selection, or MAS.

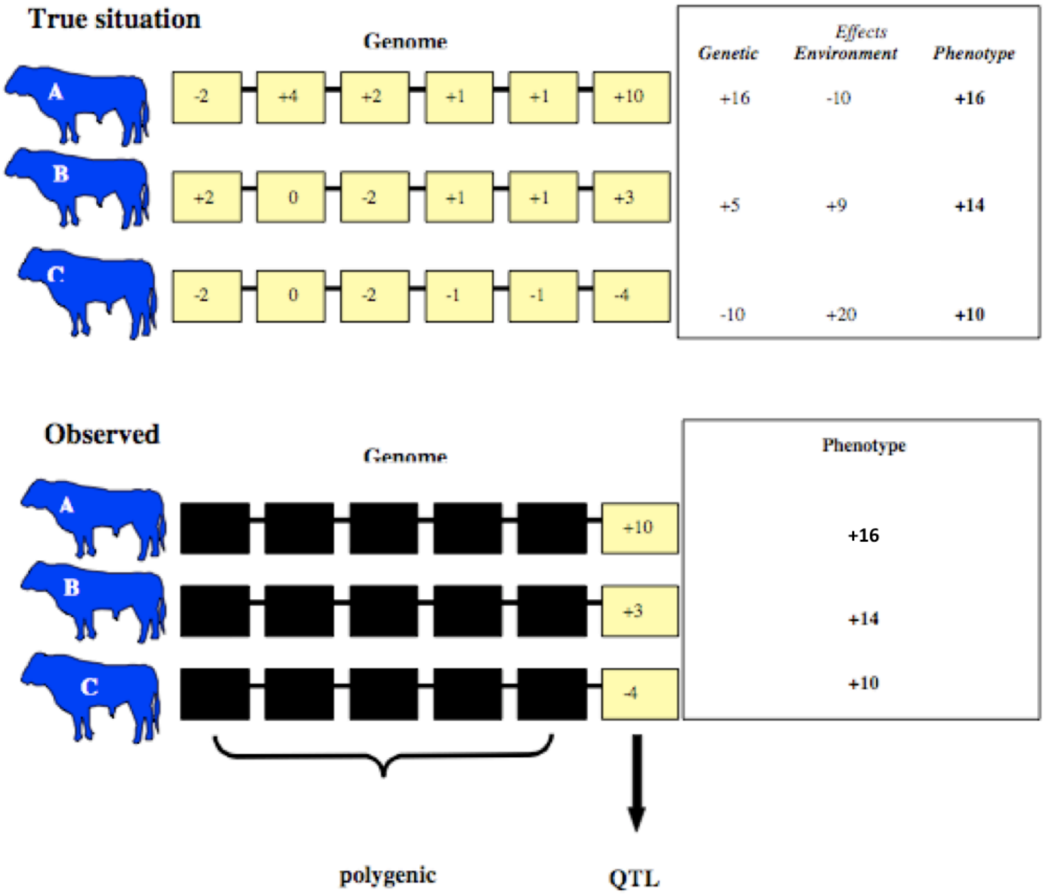


Figure 1.1: Illustration of three bulls with different phenotypes. The top drawing gives the true allelic values at the different genes affecting body weight units, which we usually cannot observe in real cases; the bottom situation illustrates what would be observed if QTL could be identified in addition to phenotype, adding significant information about the true genotype. (Figure taken from [2].)

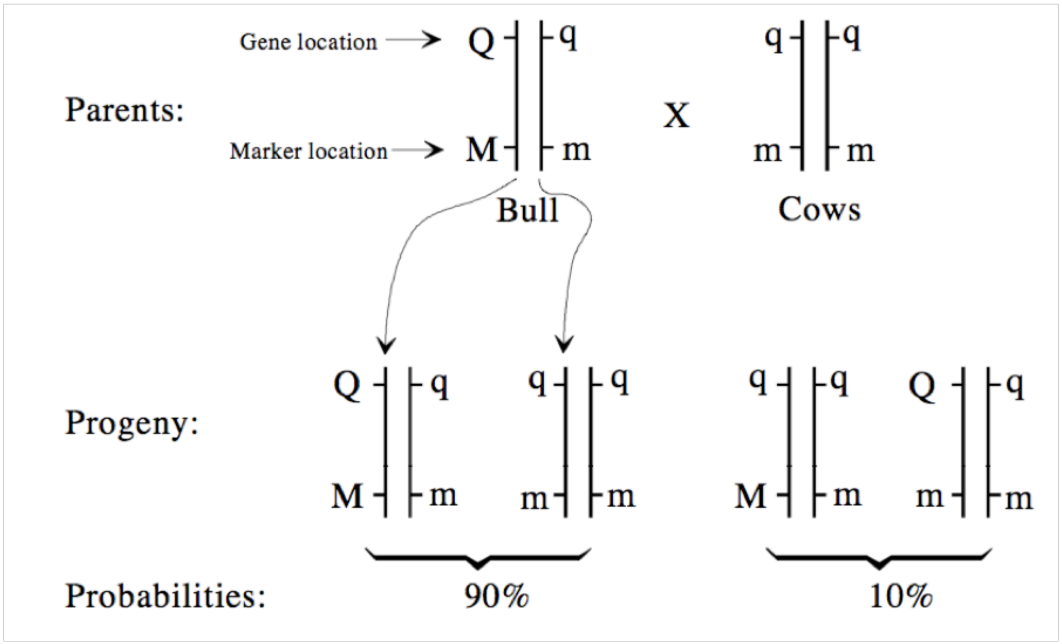
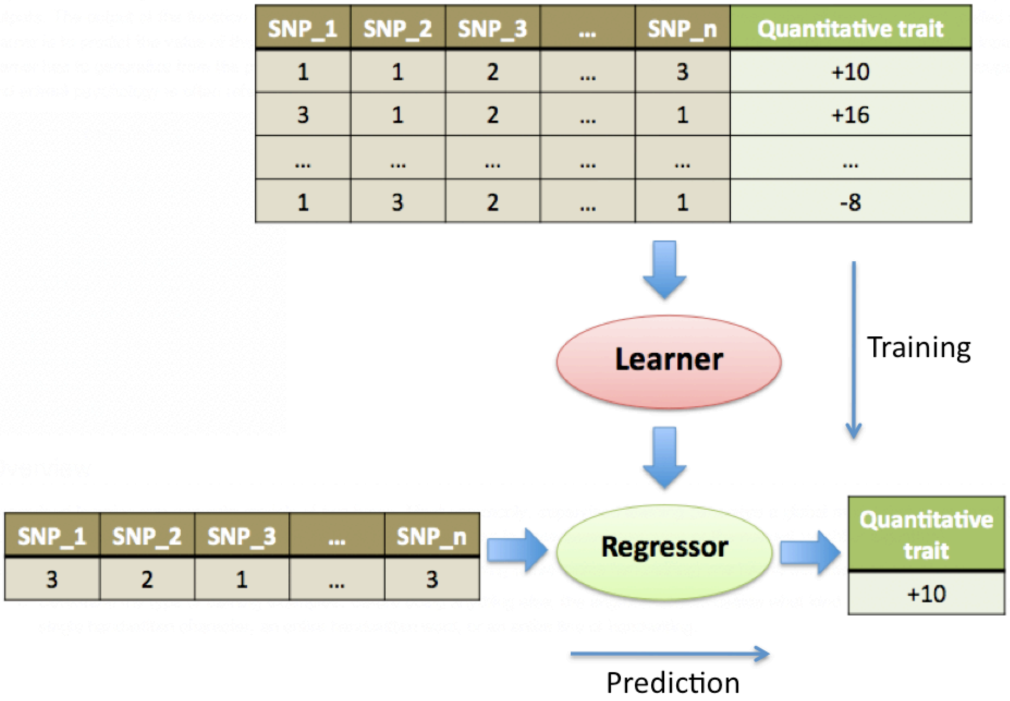


Figure 1.2: Example of the inheritance of QTL and its linked genetic marker M. Here locus Q is the location of a major genotype that affects a quantitative trait, and locus M is the location of a genetic marker. The diagram shows a pair of chromosomes for each bovine parent and its progeny. The sire is heterozygous for either locus and the dam is homozygous. For this example, we can determine for each progeny whether they received M or m allele from their sire. The recombination rate (10%) determines how often Q alleles join M alleles. (Genetic recombination refers to the process by which genetic material is broken and joined to other genetic material.) Here the recombination rate is very low, so M alleles is closely linked to Q alleles. Progeny that receive the M allele from the sire, have a high chance of having also received the Q allele, and are therefore the preferred candidates in selection. (Figure taken from [2].)

Single nucleotide polymorphisms, also known as SNPs (pronounced "snips"), are the most frequent genetic variations in the genome. A SNP is a single base substitution of one nucleotide with another, where both versions are observed in the general population at a frequency greater than 1%. With the increasing availability of affordable high-throughput SNP assays, SNPs are becoming the marker of choice in genetic analysis and are used routinely as markers in animal breeding programs [3]. For more details about SNPs, please see Section 2.2.2.

This thesis explore the task of marker-assisted selection of dairy cattle using a dense SNP map. Breeders usually take three steps to address the problem. First, the prediction of quantitative values of economically important traits are made based on the SNPs dataset. Secondly, for each cattle, the *Estimated Breeding Value* (EBV), which is the genetic merit of an animal's breeding value, is calculated based on the prediction of the traits. Finally, the selection is made based on the EBVs. As we can see, the last two steps are easy to follow if we could achieve a reasonable prediction of the traits in the first step. In this study, we only focus on the first step, *i.e.* the prediction problem. Besides, we are also interested in finding a subset of SNPs are most significantly associated with the traits, because those SNPs can be used as genetic markers to pinpoint the exact locations of the QTL. In short, the objective of this study is two-fold. The first objective is to accurately learn to predict the economically traits based on SNPs dataset. The second objective is to find a subset of SNPs that are statistically associated with the traits for QTL mapping.

In our proposed approaches, the target problem is formulated as a supervised-learning problem, where a learning algorithm takes as input a set of dairy cattle whose trait information is known with genotyped SNPs as covariates, and attempts to infer a function that will predict the traits for unseen cattle based on their SNPs; see Figure 1.3. The major challenge of this learning problem is the high dimensionality of the SNP dataset, which is a typical "large p (number of SNPs), small n (number of cattle)" problem [4]. Traditional statistical methods often lack the ability to handle such datasets with a very large p. On the other handle, the successful applications of machine-learning methods, such as Support Vector Machine (SVM), in cancer classification problems using microarray gene expression dataset (which is also "a large p small n" problem) shows good potential of machine-learning methods to handle high dimensional data [40]. We would like to see if machine-learning methods could achieve similar success when applied to the SNPs dataset.



SNP_1	SNP_2	SNP_3	...	SNP_n	Quantitative trait
1	1	2	...	3	+10
3	1	2	...	1	+16
...	...	...	...	...	...
1	3	2	...	1	-8

SNP_1	SNP_2	SNP_3	...	SNP_n
3	2	1	...	3

Quantitative trait
+10

Figure 1.3: Illustration of formulating the QTL mapping problem as a supervised-learning problem. In the top table, each row represents the genotype of a dairy cattle; each column, except the last one, represents a SNP; and the last column represents the target quantitative trait for prediction. In the supervised-learning scheme, the top table represents the training set, where the values of the prediction target, here a quantitative trait, are known. With the training set as input, the learner attempts to generate a regressor that could make good predictions on quantitative traits of future cattle. The regressor is a function that maps the input vector (here the SNPs vector) to a quantitative value. The regressor usually contains a set of parameters, and the learner uses the training set to adjust the parameters to optimize the derived regressor's performance on the target problem. This process is called training. After the training process, we then have the derived regressor, which we can use to predict the value of quantitative trait for future cattle.

The dataset used in this study comes from a diary-industry breeding program. The dataset consists of 462 dairy sires (samples) with 1341 SNPs are genotyped for each sire. Each SNP could only take 3 values, 1 (Homozygous Major), 2 (Heterozygous), and 3 (Homozygous Minor) respectively. We consider 5 prediction tasks for quantitative traits, FatEBV, FatPercentEBV, MilkEBV, ProteinEBV, and ProteinPercentEBV. The values of the quantitative traits of 157 out of the total 462 sires are withheld by the data provider as the final test set for evaluation of our methods.

Two different approaches are proposed to meet the two objectives of the study. The first approach treats the underlying genetic model of each trait as a black box, and uses all the 1341 SNPs to predict the traits without feature selection. In this approach, we only focus on the first objective, i.e. predicting the traits accurately. Two machine-learning kernel methods, Support Vector Machine (SVM) and Gaussian Process (GP), along with five statistical regression methods, Principal Component Analysis (PCA), Partial Least Square Regression (PLS), Ridge Regression, LASSO, and Elastic Net, are applied for comparison purpose.

The second approach has two steps. The first step is feature selection, which tries to find the subset of SNPs that are most significantly associated with the traits. The resulting subset of features (SNPs) is used to predict the traits in the second step. Thus the two objectives are both considered in the second approach. We compare four feature selection methods: Correlation-based feature selection, Logic Regression, M5 Prime for Linear Regression, and Haplotype Block, which makes use of some biological information to help in selecting features.

The empirical results show that the first approach generally achieves better prediction accuracies than the second approach, which means that the feature selection methods failed to increase prediction accuracy in this problem. Of the 304 samples with known traits information, the best average correlation of 5-fold cross validation by the first approach is 0.53±0.01 (by Gaussian Process), while the best average correlation by the second approach is only 0.47±0.01 (by Logic Regression + Gaussian Process).

Gaussian Process is found to be the best method, and it had the highest prediction accuracy, although the difference between GP and SVM is very small. Using 304 sires with known traits information as training data and the other 157 sires as test data, GP's average correlation of the 5 quantitative traits is 0.56±0.01. For the FatPercentEBV trait, GP's correlation reaches 0.60±0.01.

We also compare the difference between original representation and binary representation of the SNP dataset (see Section 6.3), and find that binary representation is generally the better way to represent the data. Of the 304 samples with known traits information, the average correlation of 5-fold cross validation by PLS regression using binary representation is 0.47±0.01, and the average correlation by PLS regression using original representation is only around 0.43±0.01. But for several other methods like GP, the difference is minimal.

We finally look into why feature selection methods fail in this problem. We find that the features selected based on the training data could achieve nearly perfect accuracies back on the *training* data, but they do not generalize well on the *test* data. We also find that a different cut of training and test data will often lead to a different subset of features that are selected based on the training data. We proposed several possible reasons for this to happen. Firstly, the SNP dataset might contain too much “noise”, *i.e.* quite a few SNPs might appear highly correlated with the traits by chance due to the relatively small sample size as compared with the number of SNPs, which makes the feature selection methods fail to discover the real single in the SNPs data. Secondly, it might be that none of 1341 SNPs are actually very closely associated with the traits. It is suggested that more than 30,000 SNPs are needed to cover all the possible QTL locations on the whole bovine genome [6]. Thirdly, perhaps the feature selection methods we tried are not powerful enough to detect the most relevant SNPs.

To our knowledge, this is the first time that such a high dimensional real SNP dataset (1341 SNPs) is used for breeding value estimation and QTL mapping. Although the best prediction accuracy of 0.60±0.01 is not a particularly good result, it is a very encouraging starting point. To our knowledge, this is also the first time that machine-learning methods are tried in the MAS and QTL mapping problem domain. The fact that the machine-learning kernel methods, GP and SVM, achieves higher prediction accuracy than the statistical methods fully shows the potential of machine-learning methods in handling high dimensional data. However, the feature selection methods we tried fail to identify the most relevant SNPs for this dataset. Further studies are needed to verify the applicability of feature selection methods.

In short, this dissertation shows that the machine-learning kernel methods can could achieve a fairly good prediction of quantitative traits based on a dense SNP dataset, and also the prediction performance of the kernel methods is better than that of the statistical methods tried. However, we were unable to improve the prediction accuracy with the feature selection methods we tried in this case.

The rest of the dissertation is organized as follows. In Chapter 2, we give a brief introduction to the background information of breeding value estimation and the QTL mapping problem. In Chapter 3, we provide a literature review of QTL mapping methods, as well as machine learning and statistical methods for high dimensional data. In Chapter 4 and Chapter 5, we describe seven regression methods and four feature selection methods tried in this study respectively. In Chapter 6, we present and discuss the empirical results. Chapter 7 is the conclusion of the dissertation.

Chapter 2

Backgrounds

In this section, we first give a brief introduction to the biology background knowledge that is necessary to understand the rest of our study. We then discuss two kinds of methods that are used for breeding value estimation, the Quantitative Genetic Approach and the Marker Assisted Selection (MAS). The latter makes use of the marker information of the breeding animals and is believed to be the best method for estimating future breeding value. Applying MAS could be a very challenging job because most complex traits are the result of complicated interactions among a lot of genes. How to select the markers that are most associated with the trait under selection out of hundreds, thousands, even hundreds of thousands candidate markers with only a very limited number of sample breeding animals is still an unsolved question. Finally, we introduce the machine-learning methods, which we use as an attempt to meet the challenge of MAS and answer the question of how to select the most informative markers.

2.1 Biology 101

Deoxyribonucleic acid (DNA) is the hereditary material in humans and almost all other organisms. DNA consists of two long anti-parallel strands that are made up of tiny building blocks called nucleotides. The four kinds of nucleotides that make up DNA are adenine (abbreviated as the single letter A), guanine (G), cytosine (C), and thymine (T). The DNA molecule has the shape of two intertwined spirals, referred to as a double helix.

DNA is segmented into chromosomes that are located within the nucleus of all cells. These chromosomes are the same in every cell of an organism and together make up the organisms genetic information, its *genome*. Chromosomes contain stretches of DNA called genes that code for amino acids that make proteins. Proteins are the foundation of life for all organisms, in that they are not only the major components of cell tissue, but also participate in most physical activities. The interaction and structure of proteins determine the visible characteristics or *phenotype* of an organism, while the genetic makeup of an organism is called its *genotype*.

The sequence of nucleotides that make up a gene can differ among individuals. The different forms of a gene are called *alleles*. The alleles can be the result of nucleotide differences in a gene that affect an amino acid sequence of a protein. This can result in a change, addition, or deletion of a protein that can affect the phenotype.

All organisms receive one copy of each gene from their mother and one from their father. The DNA sequence of a gene inherited from each parent may be identical, in which case the individual is said to be *homozygous* for that trait. Or the sequence of the gene from one of the parents may be different, in which case the individual is said to be *heterozygous*. Allele variations may differ in their DNA sequence by as little as a single nucleotide.

Differences among alleles caused by a single nucleotide, are called *single nucleotide polymorphisms* (SNPs). Formally, a SNP is a single base substitution of one nucleotide with another, where both versions are observed in the general population at a frequency greater than 1%. Figure 2.1 shows an example of a SNP.

SNPs can occur in both *coding* and *non-coding* regions of the genome, where only the genetic information in the coding regions is transcribed to *ribonucleic acid* (RNA) and then translated to proteins, thus genetic variations indirectly affect the phenotypes. SNPs are the most abundant source of genetic variation. For example, in the human genome, 99.9% of one individual DNA sequences will be identical to that of another person. Of the 0.1% difference, over 80% will be the SNPs, which makes SNPs of great value for biomedical research and for developing pharmaceutical products or medical diagnostics. SNPs are also evolutionarily stable, i.e. low mutation rate, making them easier to follow in population studies.

2.2 Breeding Value Estimation

Over the past 50 years, genetic improvement through artificial selection has contributed to the enormous advances in productivity that have been achieved in plant and animal species that are of agricultural importance. Selection for economically

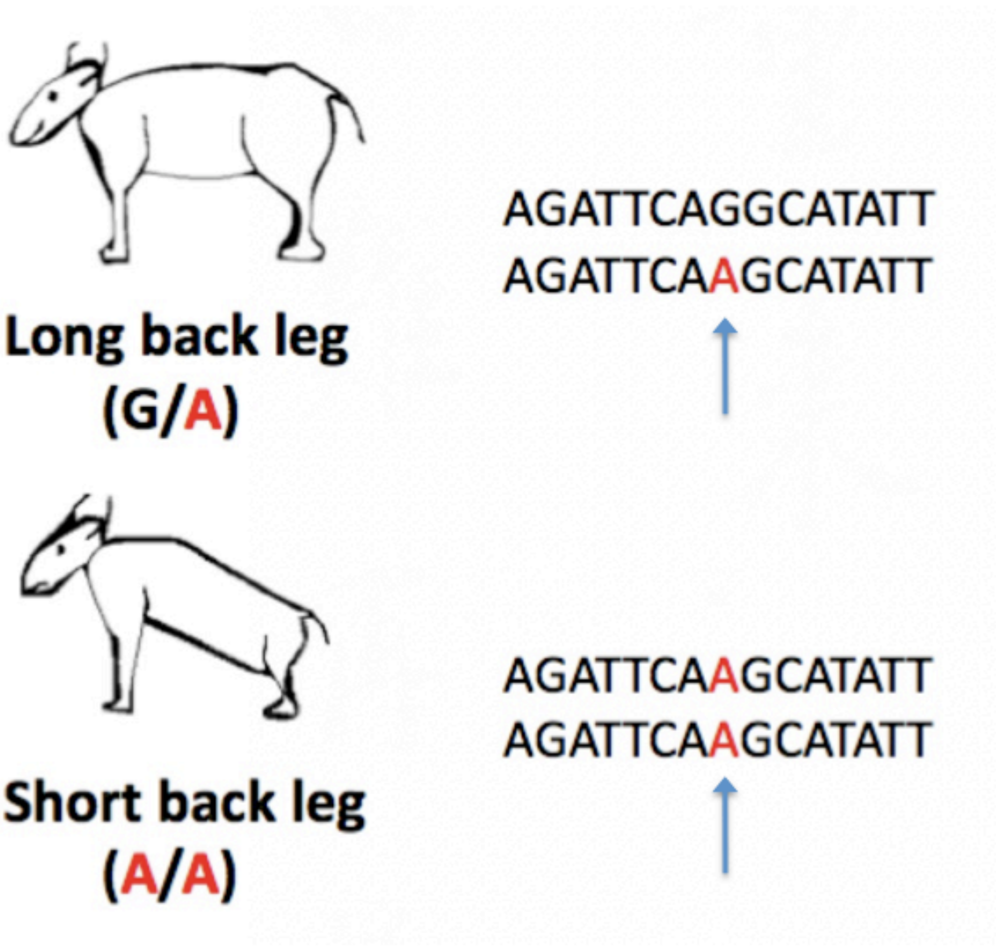


Figure 2.1: An example of a SNP. The eighth nucleotide in the DNA segment is a polymorphism with two alleles A and G. In this case, the SNP is believed to be associcated with the length of bull's back leg. Individuals with A/G genotype of this SNP are likely to have long back leg; individuals with A/A genotype are likely to have short back leg. (Figure taken from [2].)

important traits in animal and plants is traditionally on the basis of observable phenotypes, which are used to estimate the breeding values. The *Estimated Breeding Value* (EBV) of an animal is the genetic merit of that animal's genes to its progeny, which is of great interest to the breeders, as it is the basis of ongoing genetic improvement.

There are basically two types of approaches for breeding value estimation, i.e. Quantitative Genetic Approach [44] and Marker Assisted Selection (MAS) [7].

2.2.1 Quantitative Genetic Approach

Estimation of breeding value based on an animals phenotype alone can already be quite accurate for highly heritable traits. However, when animals need to be compared across herds, things get more complicated, as genetic and environmental influences have to be disentangled. To achieve this, more sophisticated statistical methods are used, leading to Best Linear Unbiased Prediction (BLUP) of breeding values [48]. Besides allowing across herd comparisons, BLUP also uses all available information about an animal's breeding value, including the animal's pedigree information. Selection accuracy is strongly dependent on the degree of data recording, which requires a range of considerations related to cost and infrastructure. In data recording, individual performances need to be related to animal identification. If BLUP is used to generate EBVs, an animals pedigree also needs to be known (in principle, for each animal only its sire and dam). If pedigree is not recorded, breeding value can be assessed on its own performance only, and could be limited to genders, *e.g.* milk production traits are only available for cows.

The phenotypic approach described above is formally called the "quantitative genetic approach". Because the phenotypic information represents a collective effect of all genes and environment, the genetic architecture of the trait itself is treated as a black box, with no knowledge of the number of genes that affect the trait, let alone of the effects of each gene or their locations in the genome. More specifically, it is based on Fishers infinitesimal genetic model, in which the trait is assumed to be determined by an infinite number of genes, each with an infinitesimally small effect [44].

The tremendous genetic improvements of the breeding animals that have been achieved attest to the usefulness of the phenotypic approach. Nevertheless, quantitative genetic selection has several limitations because the phenotype is not always a perfect predictor of the breeding value of an individual. For example, some traits are of low heritability (yield in plants), some are expensive to record (meat quality in animals), and some are only observable on one gender (milk production in dairy cattle), etc. The ideal situation for quantitative genetic selection is that the trait has high heritability and that the phenotype can be observed in all individuals before reproductive age with a relatively mild cost. This ideal is hardly ever achieved, which limits the effectiveness of quantitative genetic selection.

2.2.2 Marker Assisted Selection (MAS)

Recently high-throughput genotyping techniques have been developed, which allows the use of molecular markers as aids in genetic selection programs. This will help breeders in shifting traditional breeding to Marker Assisted Selection (MAS). Because the molecule markers of the animals could be obtained on both genders and at a young age, MAS could help to alleviate the limitations of quantitative genetic approach.

The idea behind MAS is that there may be genes with significant effects that can be targeted specifically in selection. Most traits of economic importance are quantitative traits that most likely are controlled by a fairly large number of genes. However, some of these genes might have a larger effect. Such genes can be called major genes located at Quantitative Trait Locus (QTL), which is a region of DNA that is associated with a particular phenotypic trait. As we have mentioned earlier, in practice, we rarely know the genotypes of the QTL, as the exact gene locations are often unknown. But we can use genetic markers, which are land-markers on the genome, to track the QTL.

Single nucleotide polymorphisms (SNPs) are the most abundant resource of genetic markers in the genome. For instance, Wong *et al.* [45] reported a genetic variation map of the chicken genome containing 2.8 million SNPs and demonstrated how the information can be used for targeting specific genomic regions. Likewise, Hayes *et al.* [49] found 2507 putative SNPs in the salmon genome that could be valuable for marker-assisted selection in this species. In this study, we also use SNPs as the genetic markers to improve the selection for traits of interest.

2.3 Challenges

Millions of SNPs across multiple species have been identified so far, but it is not always clear how to best use this information. It is believed that most complex traits are the result of complicated interactions among multiple genetic factors, in addition to a collection of environmental influences [8]. An important challenge that faces molecular association study in the post genomic era is to understand the inter-connections from a network of genes and their products that are initiated and mediated by a variety of environmental changes.

Traditional statistical methods often lack the ability to identify such kinds of interactions because of the inflexibility of the models and the large sample sizes required for accurate parameter estimation. For example, in a two-locus model, where we assume that there are only two loci on the genome that are closely correlated with a disease, it is a simple matter to consider the effect on that disease of all possible genotype combinations, but this quickly becomes a combinatorial challenge as the number of loci (i.e. the dimensionality of the dataset) increases. Traditional parametric statistical methods are limited in their ability to identify interacting susceptibility genes in small sample sizes because of the sparseness of the data in high dimensions. This phenomenon is referred to as the curse of dimensionality [50]; as the number of interacting genes increases, the number of genotype combinations (i.e. the dimensionality) increases exponentially, leading to the need for commensurately larger sample sizes. Unfortunately, collecting genetic datasets with such a large number of observations is prohibitively expensive. Thus, new analytical and computational methods are needed to improve the power for characterizing genetic variations that are non-redundant, through which, one can then identify the target SNPs that are most likely to affect the phenotypes.

Another drawback of traditional statistical methods for identifying interactions is the need to specify a model for the interaction. The problem is particularly acute when, again, the dimensionality, and hence the number of possible interactions, is large. Logistic regression [50], for example, models the probability of disease (p) as a logit transformation of the linear function of the independent variables. The logit transformation of p, $\ln(\frac{p}{1-p})$, is used to prevent p from taking on values outside the interval [0, 1]. The probability of having disease d given that two SNPs A, B are independent, $p(d|A, B)$, can be modeled as (Eq. 2.1):

$$p(d|A,B) = \frac{\exp(\alpha + \beta_1 A + \beta_2 B)}{1 + \exp(\alpha + \beta_1 A + \beta_2 B)} \tag{2.1}$$

where the independent variables are the polymorphisms A and B, which take on discrete genotype values corresponding to the three genotypes; exp() is the exponential function; and $\alpha \in \mathbb{R}$ and $\boldsymbol{\beta} = \{\beta_1, \beta_2\} \in \mathbb{R}^2$ are regression parameters. To model an interaction between A and B, the form of the interaction must be specified. For example, for some types of interactions between SNP A and SNP B, $p(d|A,B)$, can be modeled by inserting a product term of the form $\beta_3 AB$ into Eq. 2.1 (see Eq. 2.2):

$$p(d|A,B) = \frac{\exp(\alpha + \beta_1 A + \beta_2 B + \beta_3 AB)}{1 + \exp(\alpha + \beta_1 A + \beta_2 B + \beta_3 AB)} \tag{2.2}$$

A test of the null hypothesis of no interaction can be carried out by testing whether $\beta_3 = 0$. Rejection of this null hypothesis provides evidence for an interaction on a multiplicative scale, but the inability to reject the null hypothesis could mean that the form of the interaction requires operations more complex than simple multiplication. Using logistic regression to detect interactions when main effects are present has been investigated in [9].

One of the advantages of logistic regression is the simple physical interpretation of the model and its parameters as they relate genotypes to probability of disease. However, the advantage of interpretability is nullified if the method is unable to determine which variables interact. A framework for understanding interactions is necessary when analyzing genetic data, otherwise useful knowledge (*e.g.* gene-gene interactions) will go undetected. Machine learning offers a powerful alternative to traditional statistical methods, an alternative that generally does not require an explicit model form and is able to detect nonlinear interactions in high-dimensional datasets.

2.4 The Promises of Machine Learning Methods

Machine learning (ML) is the study and computer modeling of learning processes including the acquisition of new declarative knowledge, organization of new knowledge into general effective representations, and the discovery of new facts through observation and experimentation [12].

In this study, we focus on supervised machine learning, where the output variable guides the learning process. The goal of supervised learning is to build a classifier (or regressor) that can predict the output variable given some input variables. The output variable can be a continuous value (called regression), or a discrete class label (called classification). (As here we try to predict the quantitative traits, our problem is a regression problem.) The supervised learning problem can be formulated as follows. Given a training set,

$$\mathbb{D} = \{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_n, y_n)\}$$

where $\boldsymbol{x}_i$ is an observation with p covariates $\boldsymbol{x}_i = \{x_{1i}, \cdots, x_{pi}\}$, and y_i is the prediction target, we want to build a mapping $\mathcal{F}$,

$$\mathcal{F} : \mathbb{D} \rightarrow \mathcal{F}_{\mathbb{D}}$$

where

$$\mathcal{F}_{\mathbb{D}} : \boldsymbol{X} \rightarrow Y$$

which can be used to predict output for new observation $\boldsymbol{x}_{new}$,

$$\mathcal{F}_{\mathbb{D}}(\boldsymbol{x}_{new}) \rightarrow y_{new}$$

The performance measure of the machine learning algorithm $\mathcal{F}$ is defined by how well the resulting classifier (or regressor) $\mathcal{F}_{\mathbb{D}}$ can predict outcomes from independent test data, based on the rules it has learned from the training data.

Machine learning is based on traditional statistical models, but it is more focused on learning from experience and results in a system that can continuously self-improve, and thereby offer increased efficiency and effectiveness.

The prediction accuracy of different machine learning programs varies and depends on the type of problem, dataset and the algorithm used. Examples of application domains include protein classification [13], tissue classification for different types of cancer [14], protein secondary structure prediction [15], text mining [16], protein-protein interactions [17] and RNA binding proteins [18]. The most common ML algorithms include decision trees, production rules, support vector machines, nave Bayes, neural networks, and genetic algorithms. There are also several free suites of machine learning software, including Weka [19], C4.5 [20], and GIST [21], which makes machine learning methods available to the public.

Chapter 3

Related Works

Since the pioneering statistical work by Lander and Botstein [22], much effort has been devoted to improve the efficiency and accuracy of QTL mapping. However, several characteristics of the genomic dataset complicate the application of classical statistical methodologies. First, large amounts of missing molecular markers, due to failure in genotyping or selective genotyping, are quite common in practice. When markers are sparse, the missing genotype information between markers must be inferred. Second, the molecular markers on the same chromosome are highly correlated, which makes it difficult to identify the "major markers" that we are looking for. Third, also the biggest challenge, the number of molecular markers is usually relatively large compared to the sample size. This is called the "large p small n" problem [4].. When SNP or gene expressions from microarray experiment are used as molecular markers, the number of molecular markers (p) is even much larger than the sample size (n), *i.e.*, $n \ll p$.

In this section, we provide a brief literature review of the QTL mapping methods that are designed to tackle the challenges mentioned above, as well as the machine learning and statistical methods that have been successfully applied to the "large p, small n" problems in general.

3.1 QTL Mapping Methods Review

3.1.1 Single-QTL Model

Conventional methods for the detection of QTL are based on a comparison of single-QTL models, where we assume there is a single locus on the genome that is associated with a quantitative trait, versus a model assuming no QTL [51]. These methods are designed to detect a single QTL at a time based on a statistical test that the values of a single candidate position for a QTL has significant effect or not. The test was constructed to test each position in a genome and thus created a genome scan for QTL analysis.

Lander and Botstein [22] presented a likelihood-based framework for interval mapping (IM), where the putative QTL genotype was conditional upon a pair of flanking markers' genotypes as well as the phenotype. A least square equivalence of IM [51] was also proposed where phenotypic values were regressed onto expected genetic coefficients of a putative QTL. Motivated by the conditional independency between marker genotypes, composite interval mapping [25] proposed to introduce additional flanking markers as covariates into the likelihood function to reduce the confounding effects from nearby QTL when scanning the current interval.

Though intuitive and widely used, these methods are still insufficient to study the genetic architecture of complex quantitative traits that are affected by multiple QTLs, *i.e.* genetic variations located elsewhere on the genome could have an interfering effect. Even non-existing so-called "ghost QTL" may appear [51]. As a consequence, the power of detection may be compromised, and the estimates of locations and effects of QTLs may be biased [52]. Therefore, the multiple-QTL model, where the effects of multiple QTLs are mapped simultaneously, is proposed.

3.1.2 Multiple-QTL Model

Multiple-QTL mapping has become the state-of-the-art gene mapping procedure [26]. QTL mapping using multiple-QTL model has been viewed as a model selection issue [53]. Rather than fitting pre-specified models to the observed data, model selection approaches proceed by identifying the QTL models from a set of potential QTL models that are best supported by the data. Various model selection methods have been recently proposed for genome-wide multiple-QTL mapping from both frequentist and Bayesian perspectives.

Frequentist approaches sequentially add or delete QTL using forward and backward or stepwise selection procedures and apply criteria such as P-values or a modified Bayesian information criterion (BIC) to identify the "best multiple-QTL model". Kao *et al.* [54] adopted a stepwise regression approach to adding and deleting QTL progressively until the model is stabilized. Carlborg *et al.* [55] proposed using Genetic Algorithm to search for QTL in the genome to improve computational efficiency. The Bayesian information criterion (BIC) has been investigated by Ball

[57], Piepho and Gauch [58], Broman and Speed [53], and Bogdan *et al.* [59]. Such methods usually pick a single “good” model, ignoring the uncertainty about the model itself in the final inference [60].

Bayesian approaches for multiple-QTL mapping build on the likelihood function for the observed phenotypic and marker data, by assigning a prior probability to each model and prior distributions to the unknowns of each model. Inference is then based on the conditional distribution of the unknowns given the observed data, *i.e* the posterior distribution. The Bayesian approach can simultaneously address both model and parameter uncertainty [61].

In Bayesian analysis, Markov chain Monte Carlo (MCMC) [62] is broadly used to evaluate complex integrals to summarize posterior distributions of all relevant parameters by random sampling and simulation iteration algorithm. A recent development in MCMC methodology is the reversible jump algorithm, an extension of Gibbs sampler and Metropolis sampler, which permits posterior samples to be collected from posterior distributions with varying dimensions [63]. The reversible jump algorithm is able to generate the posterior sample of the number of QTL, the crucial parameter in QTL mapping; thereby a Bayesian inference of QTL number can be performed based on its posterior samples. Thus, the Bayesian method, incorporated with the reversible jump algorithm as well as Gibbs sampling and Metropolis samplers [64, 65, 66, 67, 68], can yield posterior densities for not only the QTL locations and the corresponding effects of a specified number of QTL but also the QTL number itself, which considerably broadens the scope of its application and is playing a more and more important role in QTL mapping. However, the practical implementation of Bayesian methods entails two major challenges: calculation of the posterior distribution and specification of the prior distributions.

There is another class of methods for handling models with a large number of model effects that require no variable selection. This class of approaches treats the genetic model as a black box and estimates the effects of all markers simultaneously without first subjecting to variable selection. The problem of high dimensionality is handled by the so-called shrinkage estimates [69, 70], where all potential model effects are included in the model but the estimated effects are forced to shrink toward zero.

Whittakeret *et al.* [71] first applied ridge regression to marker-assisted selection and showed that ridge regression can substantially improve the selection efficiency.

In their analysis, markers included in the model were selected on the basis of QTL mapping results and the number of markers was much smaller than the number of observations. Xu [67] later showed that ridge regression is not a viable choice for QTL mapping if the model includes too many markers on the genome. The reason is that having "too many" markers in the regression model produces serious colinearity, causing unstable least-squares estimates and a poor prediction of the quantitative trait. Xu [67] found that in fact most markers had negligible effects, and modified the ridge regression by allowing each marker effect to have its own variance parameter, which serves as a coefficient of penalty so that a marker with a negligible effect will have an extremely small variance that will cause its coefficient to be close to zero. A similar approach was also developed by Gianola *et al.* [72] from a marker-assisted selection perspective. The major difference between Xu [67] and Gianola *et al.* [72] is that Xu's method can estimate the QTL variance using only a single regression coefficient whereas the method of Gianola *et al.* estimates the QTL variance using a batch of regression coefficients.

Recently, several semi-parametric and non-parametric methods are also proposed for QTL mapping. Gianola *et al.* [73] discussed semi-parametric procedures analyzing complex phenotypic data involving massive genomic information. These authors argued that application of the parametric additive genetic model in selective breeding of livestock produced tangible dividends, as shown in Dekkers and Hospital [74], and proposed combining a nonparametric treatment of effects of molecular SNPs with features of the additive polygenic mode of inheritance. Gonzlez-Recio *et al.* [75] proposed a non-parametric procedure, i.e. reproducing kernel Hilbert spaces regression, for prediction of total genetic value for quantitative traits, which made use of phenotypic and genomic data simultaneously. These methods use weaker assumptions than traditional fully parametric models and allow accounting for non-additive effects without explicit modeling.

3.2 Machine learning and statistical methods for the "large p small n" problems

In this section, we review work in machine learning and statistical methods for handling the "large p small n" problems.

Like the SNP dataset, the analysis of DNA microarray data is a typical "large p, small n" problem. Microarray chips technologies allow the monitoring of gene expression levels for thousands of genes simultaneously. It is believed that such technologies may lead to a better understanding of the molecular variations. Therefore they have been increasingly applied to prediction and diagnosis of cancer. Similar to the QTL mapping problem, the major challenge of the gene expression data is the huge number of genes.

Since the advent of microarray chips, machine learning methods have been playing a pivotal role in analyzing the generated data. Quite a few machine-learning classification methods have been proposed in this problem domain, including k nearest neighbor [26], random forest [35], boosting [42], support vector machine [40], Bayesian network [41] and multi-layer perception [43]. Recently, Pirooznia *et al.* [91] compared various microarray classification methods, including SVM, RBF Neural Nets, MLP Neural Nets, Bayesian, Decision Tree and Random Forrest methods, and found that SVM had the highest classification accuracy. For other reviews and comparisons of different machine learning methods applied on microarray data, please refer to [36, 37, 38].

Some novel feature selection methods are specially designed to handle the high dimensionality of microarray/SNP dataset from the machine learning paradigm. Multifactor dimensionality reduction (MDR) has been proposed and implemented for reducing SNPs dimension [10, 11, 32, 34]. Ritchie et al. [32] reported the optimization of the architecture using genetic programming neural networks (GPNN) to detect and model gene-gene interactions in studies of human diseases. Recently, Ruczinski et al. [76] demonstrated logic regression based identification of SNP interactions for the disease status in case-control study. In comparison with some well-known classification methods such as CART [103] and Random Forest [35], logic regression showed a good classification performance when applied to SNP data. Also, genetic algorithm with K-nearest neighbor classifier (GAKNN) has been proposed to find the most relevant genes associated with tumor classification [39].

Several statistical methods have also been applied to the microarray/SNP dataset. Principal component analysis (PCA) [28] and partial least square (PLS) [29] have been proposed in an attempt to find the most significant component that explains most of the variance in the dataset. Yuan and Lin [30] proposed the group-Lars and group-Lasso methods. Zou and Hastie [79] proposed Elastic Net, which is a regulation path regression method, for modeling gene-gene interactions.

The successful applications of these methods in handling the “large p small n” problem of microarray/SNP dataset inspires us to see how well they will work in the QTL mapping problem, which has similar challenges except that the prediction target is quantitative output instead of binary, and that the values of SNPs are discrete instead of numeric.

Chapter 4

Genome Breeding Value Estimation Approaches

In this section, we will introduce seven methods that are known for their abilities to process high dimensional data in practice, so we consider using them to predict EBVs using the whole genome (all SNPs). Two kernel methods, Support Vector Machine (SVM) and Gaussian Process (GP), from the machine-learning paradigm along with five statistical methods are included for comparison. The five statistical methods can be further divided into two categories, dimension transformation methods, which include Principal Component Analysis regression (PCA) and Partial Least Square regression (PLS), and regularization methods, which include Ridge Regression, LASSO, and Elastic Net.

4.1 Support Vector Machine (SVM)

The support vector machine (SVM) algorithm is originally proposed as a classification algorithm [46, 47] that provides state-of-the-art performance in a wide variety of application domains, from pattern recognition problems to computational biology, including handwriting recognition, face detection, text categorization, microarray gene expression analysis and prediction of protein-protein interactions.

4.1.1 Optimal Hyperplane for Linearly Separable Patterns

In the classification case, we try to find an optimal hyperplane that separates two classes. We are given some training data, a set of points of the form.

$$\mathbb{D} = \{(\boldsymbol{x}_i, c_i) | \boldsymbol{x}_i \in \mathbb{R}^p, c_i \in \{-1, 1\}\}_{i=1}^{n}$$

where the c_i is either $+1$ or -1, indicating the class to which the point belongs, and $\boldsymbol{x}_i$ is a p-dimensional input vector. We assume that the class represented by the subset $\{\boldsymbol{x}_i | c_i = +1\}$ and the class represented by the subset $\{\boldsymbol{x}_i | c_i = -1\}$ are *linearly separable.* The decision surface equation in the form of a hyperplane that separates the data is

$$\boldsymbol{w}^T \boldsymbol{x} + b = 0 \tag{4.1}$$

where $\boldsymbol{w} \in \mathbb{R}^p$ is an adjustable weight vector, and $b \in \mathbb{R}$ is a bias.

We can thus write:

$$\begin{cases} \boldsymbol{w}^T \boldsymbol{x} + b \geq 0 & for \;\; c_i = +1 \\ \boldsymbol{w}^T \boldsymbol{x} + b < 0 & for \;\; c_i = -1 \end{cases}$$

which can be written as:

$$c_i(\boldsymbol{w}^T \boldsymbol{x} + b) \geq 0 \tag{4.2}$$

But we do not simply want the instances to be on the correct side of the hyperplane, but we also want them some distance away, for better generalization. For a given weight vector $\boldsymbol{w}$ and bias b, the separation between the hyperplane and the closest data point is called the *margin of separation.* The goal of a support vector machine is to find the particular hyperplane for which the margin of separation is maximized. Under this condition, the decision surface is referred to as the *optimal hyperplane*; see Figure 4.1.

Rather than meeting the constraint of Eq. 4.2, we instead want to find $\boldsymbol{w}$ and b such that

$$\begin{cases} \boldsymbol{w}^T \boldsymbol{x} + b \geq +1 & for \;\; c_i = +1 \\ \boldsymbol{w}^T \boldsymbol{x} + b \leq -1 & for \;\; c_i = -1 \end{cases}$$

which can be written as:

$$c_i(\boldsymbol{w}^T \boldsymbol{x} + b) \geq +1 \tag{4.3}$$

The particular data points $(\boldsymbol{x}_i,\, c_i)$ satisfying the equality sign,

$$\begin{cases} \boldsymbol{w}^T \boldsymbol{x}_i + b = +1 & for \;\; c_i = +1 \\ \boldsymbol{w}^T \boldsymbol{x}_i + b = -1 & for \;\; c_i = -1 \end{cases}$$

are called support vectors.

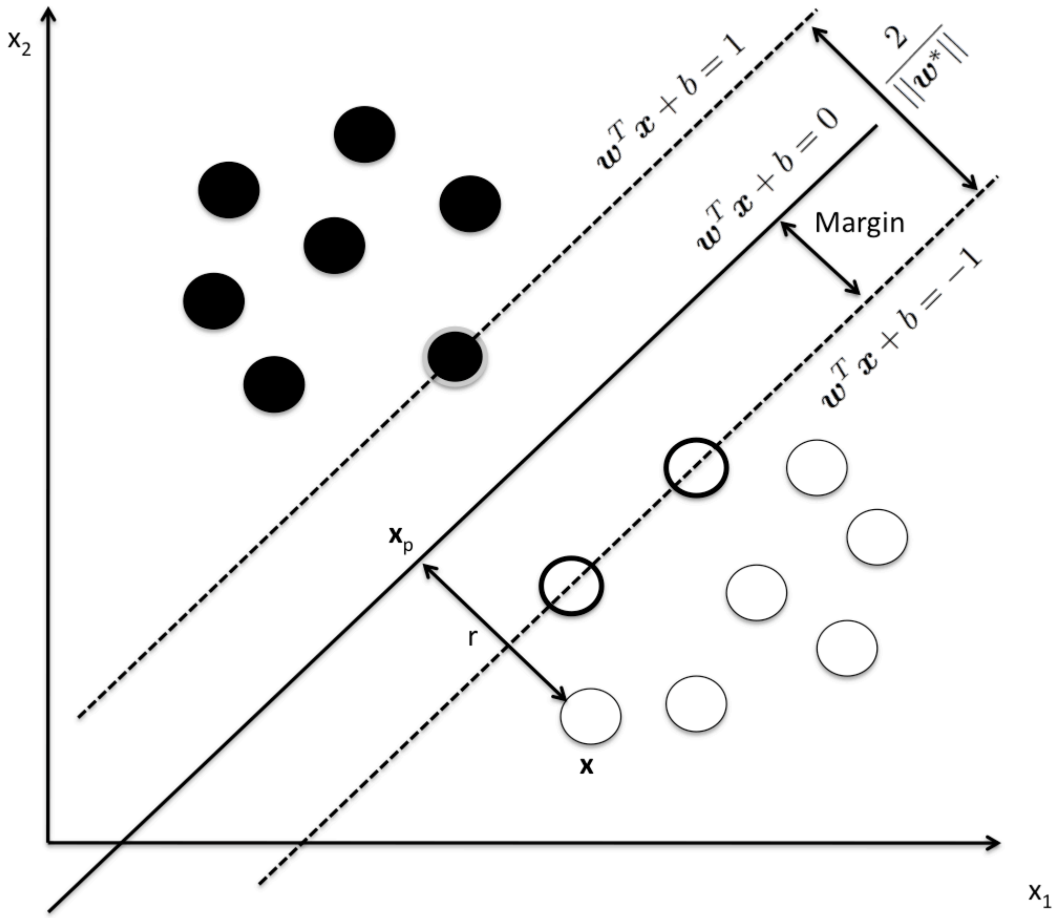

x_2
$\boldsymbol{w}^T\boldsymbol{x}+b=1$
$\frac{2}{||\boldsymbol{w}^*||}$
$\boldsymbol{w}^T\boldsymbol{x}+b=0$
Margin
$\boldsymbol{w}^T\boldsymbol{x}+b=-1$
$\mathbf{x}_p$
r
$\mathbf{x}$
x_1

Figure 4.1: Operating mode of a Support Vector Machine in linearly separable case. The SVM algorithm seeks to maximize the margin around a hyperplane that separates members of the positive class (marked by white circles) from members of the negative class (marked by black circles). Only support vectors (circles on the dotted lines) are required to define the optimally defined hyperplane. The distance between the support vectors and the hyperplane is called the margin. The optimal hyperplane is found when the margin is maximized.

Let $\boldsymbol{w}^*$ and b^* denote the optimum value of the weight vector and bias. Optimal hyperplane is defined as

$$(\boldsymbol{w}^*)^T\boldsymbol{x} + b^* = 0$$

The discriminant function

$$g(\boldsymbol{x}) = (\boldsymbol{w}^*)^T\boldsymbol{x} + b^*$$

gives an Euclidean measure of the distance from $\boldsymbol{x}$ to the optimal hyperplane. In order to see this, we express $\boldsymbol{x}$ as

$$\boldsymbol{x} = \boldsymbol{x}_p + r\boldsymbol{w}^*/||\boldsymbol{w}^*||$$

where $\boldsymbol{x}_p$ is the normal projection of $\boldsymbol{x}$ onto the optimal hyperplane, and r is the desired Euclidean distance.

By definition $g(\boldsymbol{x}_p) = 0$, it follows that

$$g(\boldsymbol{x}) = (\boldsymbol{w}^*)^T\boldsymbol{x} + b^* = r||\boldsymbol{w}^*||$$

$$r = g(\boldsymbol{x})/||\boldsymbol{w}^*||$$

For any support vector $\boldsymbol{x}_s$, the Euclidean distance from the support vector $\boldsymbol{x}_s$ to the optimal hyperplane is

$$\begin{cases} r = 1/||\boldsymbol{w}^*|| & if \;\; c_s = +1 \\ r = -1/||\boldsymbol{w}^*|| & if \;\; c_s = -1 \end{cases}$$

The margin of separation between the two classes is

$$\rho = 2r = \frac{2}{||\boldsymbol{w}^*||}$$

Maximizing the margin of separation between classes is equivalent to minimizing the Euclidean norm, i.e., $||\boldsymbol{w}^*||$, of the weight vector $\boldsymbol{w}$.

Formally, the constrained optimization problem may be stated as: Given the training sample $\mathbb{D} = \{(\boldsymbol{x}_i, c_i)|\boldsymbol{x}_i \in \mathbb{R}^p, c_i \in \{-1, 1\}\}_{i=1}^n$, find the optimum values of the weight vector $\boldsymbol{w}$ and bias b such that they satisfy the constraints

$$c_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 \qquad for \;\; i = 1, 2, ..., n$$

and the weight vector $\boldsymbol{w}$ minimizes the cost function

$$\Phi(\boldsymbol{w}) = \frac{1}{2} \cdot \boldsymbol{w}^T\boldsymbol{w}$$

The scaling factor 1/2 is included for convenience of presentation. This constrained optimization problem is called the *primal problem*, and can be solved by using the method of Lagrange multipliers. For more details on this, please refer to [93].

4.1.2 Optimal Hyperplane for Non-separable Classes

Now we consider the more difficult case of non-separable classes. Given such a set of training data, it is NOT possible to construct a separating hyperplane without encountering classification errors. Nevertheless, we would like to find an optimal hyperplane that minimizes the probability of classification error averaged over the training set.

The margin of separation between classes is said to be soft if a data point violates the following condition:

$$c_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1 \qquad for \;\; i = 1, 2, ..., n$$

We introduce a new set of nonnegative scalar variables $\{\xi\}_{i=1}^{n}$, called *slack variables*, into the definition of the separating hyperplane (i.e., decision surface) as follows.

$$c_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1 - \xi_i \qquad for \;\; i = 1, 2, ..., n$$

The $\{\xi\}_{i=1}$ measure the deviation of a data point from the ideal condition of pattern separability. For $0 \leq \xi_i \leq 1$, the data point falls inside the region of separation but on the right side of the decision surface. For $\xi_i > 1$, it falls on the wrong side of the separating hyperplane, see Figure 4.2.

To make the optimization problem mathematically tractable, we approximate the cost function as follows.

$$\Phi(\boldsymbol{w}, \boldsymbol{\xi}) = 1/2 \cdot \boldsymbol{w}^T \boldsymbol{w} + C \sum_{i=1}^{n} \xi_i$$

The second term $\sum_{i=1}^{n} \xi_i$ is an upper bound on the number of test errors. The parameter C controls the tradeoff between complexity of the machine and the number of non-separable points; it may therefore be viewed as a form of a “regularization” parameter. The parameter C is typically selected by the user.

We therefore have the primal problem for the non-separable case: Given the training sample $\mathbb{D} = \{(\boldsymbol{x}_i, c_i) | \boldsymbol{x}_i \in \mathbb{R}^p, c_i \in \{-1, 1\}\}_{i=1}^{n}$, find the optimum weight vector $\boldsymbol{w}$ and bias b that satisfy the constraints

$$c_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1 - \xi_i \qquad for \;\; i = 1, 2, ..., n$$

and that minimizes the cost function

$$\Phi(\boldsymbol{w}, \boldsymbol{\xi}) = 1/2 \cdot \boldsymbol{w}^T \boldsymbol{w} + C \sum_{i=1}^{n} \xi_i$$

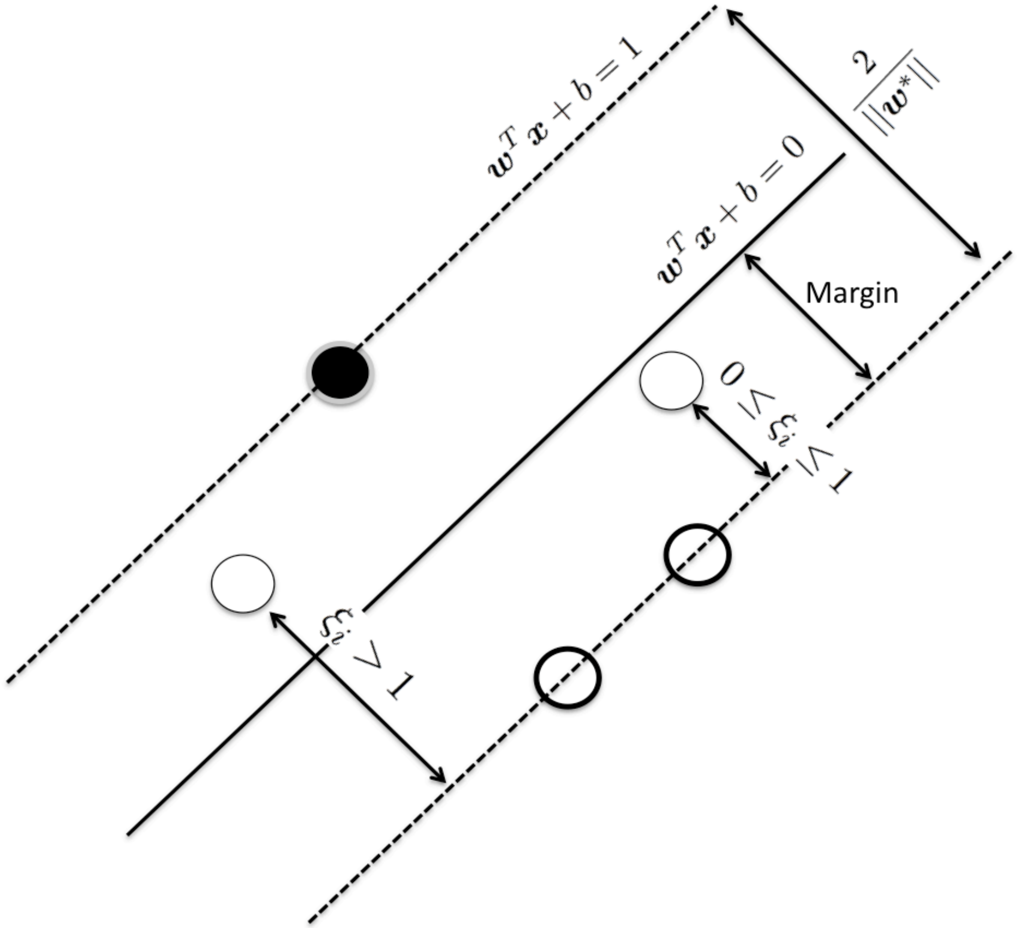


Figure 4.2: Illustration of soft-margin SVM with the introduction of slack variables ξ_i, where some data points could be fall within the margin ($0 \le \xi_i \le 1$), even on the other side of the separating hyperplane ($\xi_i > 1$).

where C is a user-specified positive parameter.

Another way to deal with the non-separable classes is to transform the original input space into a higher dimensional feature space, and then seek classes that can be linearly separable in the new space. We then can try to find the maximum-margin hyperplane in that space. This approach is called the *kernel trick*. The resulting algorithm to fit the maximum-margin hyperplane in the transformed feature space is similar, except that every dot product $\boldsymbol{x} \cdot \boldsymbol{x'}$ used in solving the linear SVM is replaced by a non-linear kernel function $k(\boldsymbol{x}, \boldsymbol{x'})$. Some common kernel functions include,

- Polynomial (homogeneous): $k_{poly,d}(\boldsymbol{x}, \boldsymbol{x'}) = (\boldsymbol{x} \cdot \boldsymbol{x'})^d$ ($d = 1$ corresponds to standard dot product.)
- Polynomial (inhomogeneous): $k_{inpoly,d}(\boldsymbol{x}, \boldsymbol{x'}) = (\boldsymbol{x} \cdot \boldsymbol{x'} + 1)^d$
- Radial Basis Function: $k_{rbf}(\boldsymbol{x}, \boldsymbol{x'}) = exp(-\gamma||\boldsymbol{x} - \boldsymbol{x'}||^2)$, for $\gamma > 0$
- Gaussian Radial basis function: $k_g(\boldsymbol{x}, \boldsymbol{x'}) = exp(-\frac{||\boldsymbol{x}-\boldsymbol{x'}||^2}{2\sigma^2})$

4.1.3 Support Vector Machine for Regression

The principle of support vector machine could be extended easily to the task of regression problems by introducing an alternative loss function [82]. Given a training set

$$\mathbb{D} = \{(\boldsymbol{x}_i, y_i) | \boldsymbol{x}_i \in \mathbb{R}^p, y_i \in \mathbb{R}\}_{i=1}^n$$

where $\boldsymbol{x}_i$ is a p-dimensional input vector, and y_i is the response variable, we want to estimate the following linear regression:

$$f(x) = (\boldsymbol{w} \cdot \boldsymbol{x} + b), \qquad \boldsymbol{w} \in \mathbb{R}^p, b \in \mathbb{R} \tag{4.4}$$

Here we consider SVR with Vapniks ϵ-insensitive loss function [47] defined as:

$$L_\epsilon(y, f(\boldsymbol{x})) = \left\{ \begin{array}{cc} 0 & |y - f(\boldsymbol{x})| \leq \epsilon \\ |y - f(\boldsymbol{x})| - \epsilon & |y - f(\boldsymbol{x})| > \epsilon \end{array} \right. \tag{4.5}$$

$$L_\epsilon = |y - f(\boldsymbol{x})| - \epsilon \tag{4.6}$$

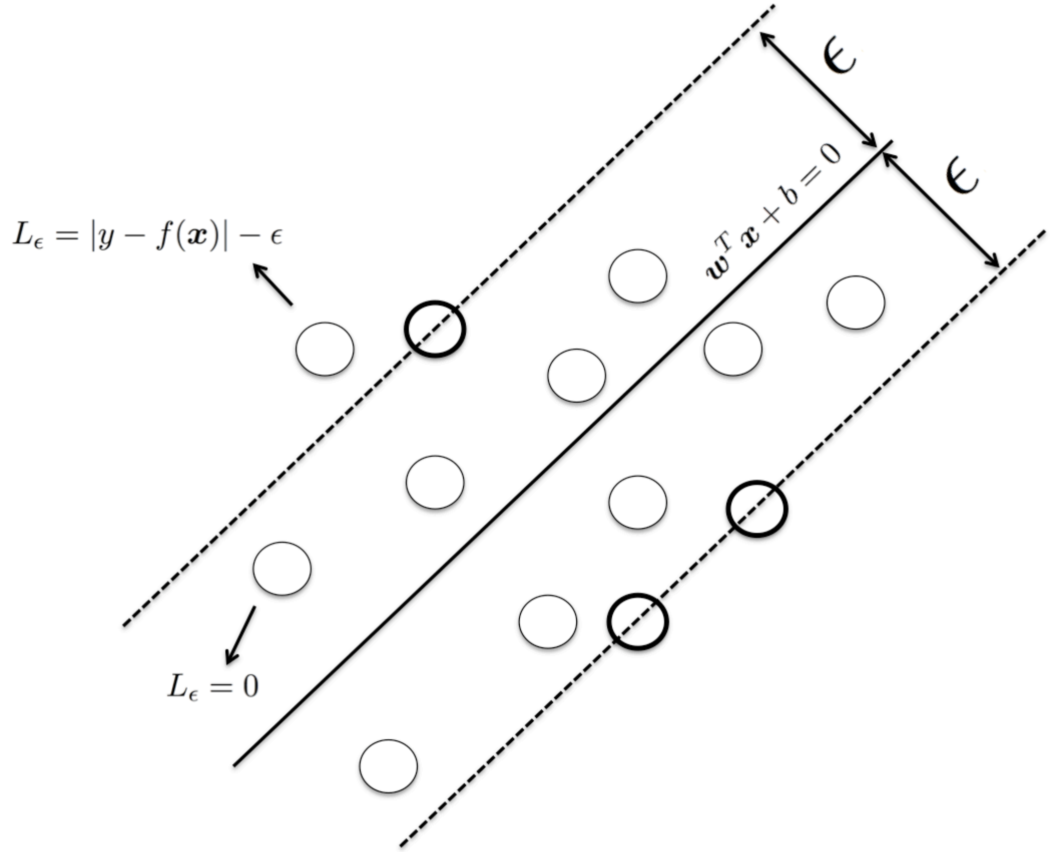

$L_\epsilon = |y - f(\boldsymbol{x})| - \epsilon$
ϵ
ϵ
$\boldsymbol{w}^T \boldsymbol{x} + b = 0$
$L_\epsilon = 0$

Figure 4.3: Illustration of support vector regression (SVR) with ϵ-insensitive loss function. All data points within ϵ distance from the regression line will have no penalty.

$$L_\epsilon = 0 \tag{4.7}$$

With the ϵ-insensitive loss function, our goal is to find a function $f(\boldsymbol{x}_i)$ that deviates at most ϵ from the actually obtained targets y_i; see Figure 4.3 In other words, we do not care about the errors as long as they are less than ϵ. The best line is defined to be that line that minimizes the following cost function:

$$R(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{l} L_\epsilon(f(y_i, \boldsymbol{x}_i)) \tag{4.8}$$

where C is again the user-specified positive parameter determining the tradeoff between the training errors and the model complexity.

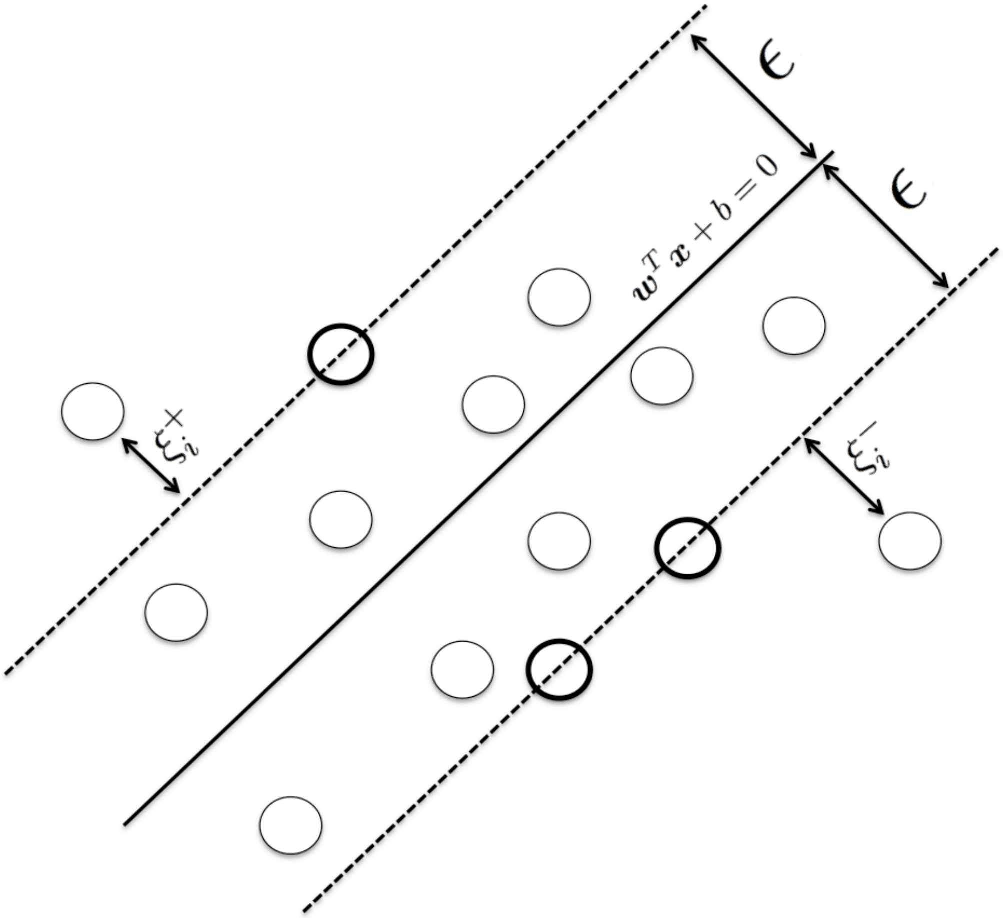


Figure 4.4: Illustration of introducing slack variables ξ^+, ξ^- to SVR only when $|f(\boldsymbol{x}_i) - y| > \epsilon$

Sometimes, however, we might not be able to find such a function f that can place all pairs $(\boldsymbol{x}, y)$ within ϵ precision, *i.e.* $|f(\boldsymbol{x}_i) - y| \leq \epsilon$. Analogous to the non-separable classes, we can also introduce slack variables ξ^{+}, ξ^{-} to cope with this problem. If the observed point is above the hyperplane, ξ_i^{+} is the positive difference between the observed value and $y + \epsilon$. Similar, if the observed point is below the hyperplane, ξ_i^{-} is the negative difference between the observed value and $y - \epsilon$; see Figure 4.4. This corresponds to a constrained optimization problem, to minimize:

$$\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{l}(\xi_i^{+} + \xi_i^{-}) \tag{4.9}$$

subject to:

$$\begin{aligned} y_i - (\boldsymbol{w} \cdot \boldsymbol{x}_i) - b &\leq \epsilon + \xi_i^{+} \\ (\boldsymbol{w} \cdot \boldsymbol{x}_i) + b - y_i &\leq \epsilon + \xi_i^{-} \\ \xi_i^{+}, \xi_i^{-} &\geq 0 \end{aligned}$$

We can also use the kernel trick in support vector regression (SVR), *i.e.* mapping the input data $\boldsymbol{x}$ into a higher dimensional feature space $\mathcal{F}$ via a non-linear mapping ϕ and then a linear regression problem is obtained and solved in this feature space.

To generalize to non-linear regression, one can again use the kernel trick to map the input data $\boldsymbol{x}$ into a higher dimensional feature space $\mathcal{F}$ via a non-linear mapping ϕ and then a linear regression problem is obtained and solved in this feature space.

4.2 Gaussian Process

Just like Gaussian distributions define the distributions over a vector of random variables, Gaussian process defines the distributions over functions, which has a formal definition as follows,

Definition 1: *A Gaussian Process is a collection of random variables, any finite subset of which have (consistent) joint Gaussian distributions.*

A Gaussian process is fully specified by its mean function $m(\boldsymbol{x})$ and covariance function $k(\boldsymbol{x}, \boldsymbol{x'})$, which is also called the kernel function. This is a natural generalization of the Gaussian distribution whose mean and covariance is a vector and matrix, respectively. We will write

$$f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x'})) \tag{4.10}$$

to represent a function f that is distributed as a GP with mean function $m(\boldsymbol{x}$ and covariance function $k(\boldsymbol{x}, \boldsymbol{x'})$, where $\boldsymbol{x}$ and $\boldsymbol{x'}$ are two vectors.

Although the generalization from distribution to process is straight forward, there might be a little confusion with the indexing. In the Gaussian distribution, the individual random variables in a vector are indexed by their positions in the vector. For the Gaussian process, it is the argument $\boldsymbol{x}$ of the random function $f(\boldsymbol{x})$ that plays the role of indexing: for every input $\boldsymbol{x}$ there is an associated random variable $f(\boldsymbol{x})$, which is the value of the function f at that location. For notation convenience, we will usually enumerate the $\boldsymbol{x}$ values of interest by the natural numbers, like $\boldsymbol{x}_i$, $f(\boldsymbol{x}_i)$. But they are not the index of the process.

Given the mean function $m((\boldsymbol{x})$ and covariance function $k(\boldsymbol{x}, \boldsymbol{x'})$, we could define distributions over functions using GPs. This GP will be used as prior for Bayesian inference. We can update this prior in the light of the training data, which gives us the posterior distribution over functions. Then we can use the posterior to make predictions for unseen test cases. More specifically, let f_T be the known function values of the training cases, and let f_P be the set of function values corresponding to the test set inputs, $\boldsymbol{x}^*$. The joint distribution of f_T and f_P is as follows,

$$\left[\begin{array}{c} f_T \\ f_P \end{array}\right] \sim \mathcal{N}\left(\left[\begin{array}{c} \mu \\ \mu^* \end{array}\right], \left[\begin{array}{cc} \Sigma & \Sigma_* \\ \Sigma_*^T & \Sigma_{**} \end{array}\right]\right) \tag{4.11}$$

where $\mu = m(x_i), i = 1, \cdots, n$ for the training means and analogously for the test means μ^*; Σ is the training set covariances, Σ_* is the training-test set covariances and Σ_{**} is the test set covariances. Since we know the values for the training set f_T, we are interested in the conditional distribution of f_P, which is expressed as

$$f_P|f_T \sim \mathcal{N}(\mu^* + \Sigma_*^T \Sigma^{-1}(f_T - \mu), \Sigma_{**} - \Sigma^{-1}\Sigma_*) \tag{4.12}$$

This is the posterior distribution for a specific set of test cases. It can be verified that the corresponding posterior Gaussian process is:

$$f|\mathbb{D} \sim \mathcal{GP}(m_{\mathbb{D}}, k_{\mathbb{D}}) \tag{4.13}$$

where

$$\begin{gathered} m_{\mathbb{D}}(x) = m(x) + \Sigma(\mathbf{X}, x)^T \Sigma^{-1}(f - m) \\ k_{\mathbb{D}}(x, x') = k(x, x') - \Sigma(\mathbf{X}, x)^T \Sigma^{-1} \Sigma(\mathbf{X}, x) \end{gathered}$$

where $\Sigma(\mathbf{X}, x)$ is a vector of covariances between every training case in the training set $\mathbf{X}$ and a test case x. These are the central equations for Gaussian process predictions. Lets examine these equations for the posterior mean and covariance.

Notice that the posterior variance $k_{\mathbb{D}}(x, x') = k(x, x')$ is equal to the prior variance $k(x, x')$ minus a positive term that depends on the training inputs; thus the posterior variance is always smaller than the prior variance, since the data has given us some additional information.

From the posterior process, we could sample function values for the test data points. Then we could either use the mean values as our predictions, or express our uncertainty of the predictions by confidence intervals.

One issue about GP is how to choose the prior mean and covariance functions. If we have enough prior information about a dataset, we could choose the prior functions to reflect the prior knowledge. But the availability of such detailed prior information is not the typical case. In order for the GP techniques to be of value in practice, we must be able to chose between different mean and covariance functions making use of the training data. With typically vague prior information, we use a hierarchical prior, where the mean and covariance functions are parameterized in terms of hyper-parameters. For example, consider the Gaussian process given by:

$$\begin{gathered} f \sim \mathcal{GP}(m, k) \\ m(x) = ax^2 + bx + c, \ \ and \ \ k(x, x') = \sigma_1^2 exp(-\tfrac{(x-x')^2}{2l^2}) + \sigma_2^2 \end{gathered} \tag{4.14}$$

where we have introduced the hyper-parameters $\boldsymbol{\theta} = \{a, b, c, \sigma_1, \sigma_2, l\}$. In order to make inference on the hyper-parameters with the training data, we compute the log likelihood of the data given the hyper-parameters $p(y|\boldsymbol{x}, \boldsymbol{\theta})$, and find the values of the hyper-parameters which maximize the log likelihood based on its partial derivatives. For more details, please refer to Rasmussen *et al.* [81].

4.3 Regularization Methods

Regularization methods, are used to prevent overfitting in statistics and machine learning problems. In statistics, overfitting is usually caused by fitting a statistical model that has too many parameters *w.r.t.* the size of the training sample. It occurs when the degrees of freedom in parameter selection exceed the information content of the data. This leads to a false model that may fit perfectly to existing data, but does not generalize well beyond the fitting data.

For example, multi-collinearity among the regressors, which means that there is redundancy in representing information, often leads to overfitting models. Given a training set $\mathbb{D}$,

$$\mathbb{D} = \{(\boldsymbol{x}_i, y_i) | \boldsymbol{x}_i \in \mathbb{R}^p, y_i \in \mathbb{R}\}_{i=1}^n$$

where $\boldsymbol{x}_i$ is a p-dimensional input vector, and y_i is the associated target, we want to estimate the following multivariate linear regression model (written in matrix form):

$$\mathbf{y} = \boldsymbol{\beta} \cdot \mathbf{X} + \epsilon \tag{4.15}$$

where $\mathbf{X}$ is the $n \times (p+1)$ matrix whose rows each representing an input vector (with a 1 in the first position to include bias in the matrix), $\mathbf{y}$ is the column vector of length n representing the regression target, and $\boldsymbol{\beta}$ is the row vector of length $(p+1)$ representing the coefficients we want to solve for in the linear model. We fit the model by least-squares estimation to obtain solutions for $\boldsymbol{\beta}$,

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{4.16}$$

By the Gauss-Markov theorem, the least-squares estimates of the coefficients have the minimum variance among all linear unbiased estimates [50]. Unfortunately, this does not necessarily mean that the least-squares estimates yield the best fit to unseen test data. When some of the regressors are (near) multi-collinear, – that is when there are linear combinations among them that show little variation –, the matrix $\mathbf{X}^T\mathbf{X}$ in Eq. 4.16 will be (nearly) singular. So the variance of $\boldsymbol{\beta}^*$, $var(\boldsymbol{\beta}^*) = E[(\beta^* - \overline{\beta^*})^2]$, will have very large elements. Correspondingly, the components of $\boldsymbol{\beta}^*$ may show unrealistically large values. Under exact collinearity, $\boldsymbol{\beta}^*$ is not even uniquely defined. In these situations, it pays substantially to use regularization methods that trade bias for variance. By adding penalties on the coefficients, estimates of $\boldsymbol{\beta}^*$ are more realistic.

Next, we talk about three regularization methods, *i.e.* ridge regression, LASSO, elastic net. The only difference between the three methods is the different penalties added on the coefficients.

4.3.1 Ridge Regression

Ridge regression [50] shrinks the regression coefficients by imposing a penalty of the sum of squares of coefficients, which is called the L2 penalty:

$$\hat{\boldsymbol{\beta}}^{ridge} = argmin\left\{\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}(\mathbf{X}_{ij}\beta_j))^2 + \lambda\sum_{j=1}^{p}\beta_j^2\right\} \tag{4.17}$$

where x_{ij} is the ith row/jth column element of a $n \times p$ matrix whose rows each represents an input vector, and y_i is the associated target.

Here $\lambda \geq 0$ is a complexity parameter that controls the amount of shrinkage. The larger the value of λ is, the greater the shrinkage is. When $\lambda = 0$, there is no penalty; when λ is very large (*e.g.* $\lambda \gg 10000$), all coefficients shrinks to 0.

Bayesian interpretation of Ridge Regression

Consider a linear model

$$y = \boldsymbol{x}^T \boldsymbol{\beta} + \epsilon$$

where $\epsilon \sim N(0, \sigma^2)$. Consider further a Bayesian approach to estimation of the p dimensional parameter vector $\boldsymbol{\beta}$ where a prior Gaussian distribution

$$\boldsymbol{\beta} \sim N(\mathbf{0}, \tau^2 \boldsymbol{I})$$

where $\mathbf{0}$ is a p-dimensional vector containing zeros and $\boldsymbol{I}$ is the identity matrix of dimension $p \times p$. Assume we have observed $D = \{((x)_i, y_i)\}_{i=1}^N$. By applying Bayes theorem, we have the posterior distribution of $\boldsymbol{\beta}$

$$p(\boldsymbol{\beta}|D) \propto exp\left\{-\frac{1}{2\sigma^2}(y_i - \boldsymbol{x}_i^T\boldsymbol{\beta})^2 - \frac{1}{2\tau^2}\sum_{j=1}^{p}\boldsymbol{\beta}_j^2\right\} \tag{4.18}$$

By maximizing $p(\boldsymbol{\beta}|D)$, we have the Bayesian estimate of $\boldsymbol{\beta}$, which turns out to be just the ridge regression estimate, where the complexity parameter λ is given by

$$\lambda = \frac{\sigma^2}{\tau^2}$$

.

4.3.2 LASSO

LASSO [70] is another regularization method, differing slightly from ridge regression. The LASSO estimate is defined by

$$\hat{\boldsymbol{\beta}}^{LASSO} = argmin\left\{\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}(x_{ij}\beta_j))^2 + \lambda\sum_{j=1}^{p}|\beta_j|\right\} \tag{4.19}$$

LASSO replaces ridge's L2 penalty $\sum_{j=1}^{p}(x_{ij}\beta_j)^2$ with the L1 penalty $\lambda\sum_{j=1}^{p}|\beta_j|$. Again, $\lambda \geq 0$ is the complexity parameter controlling the shrinkage amount. From Figure 4.5, it is easy to see when the value of λ becomes larger, some of the coefficients will shrink to zero faster under the L1 penalty than the L2 penalty. Thus when regulating the coefficients, LASSO also performs variable selection as a side effect.

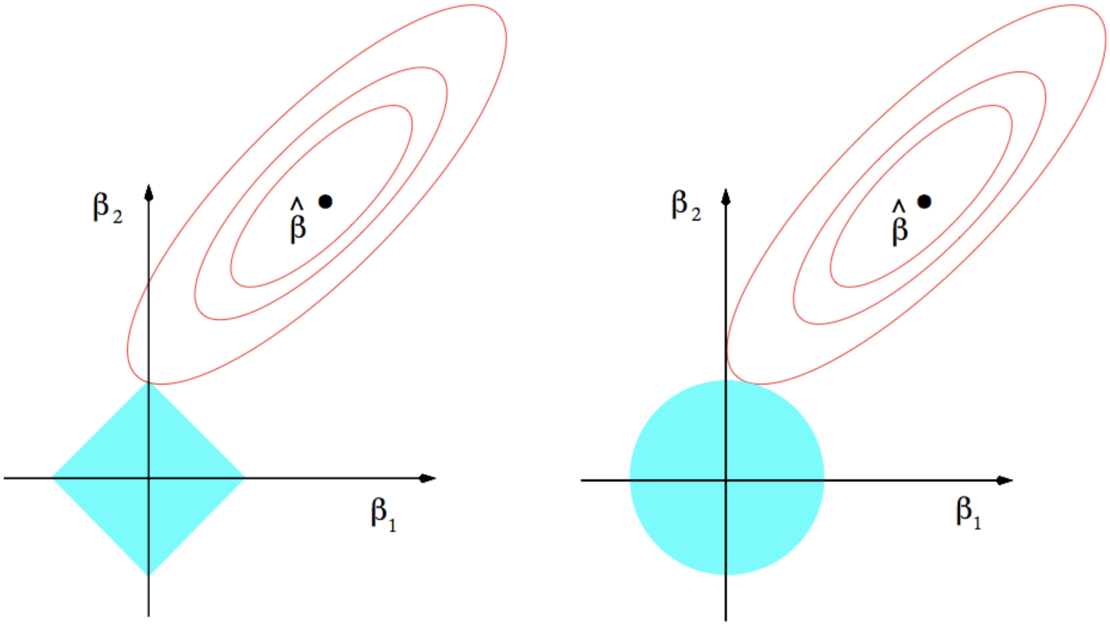


Figure 4.5: Comparison between the lasso (left) and ridge regression (right) for $p = 2$ case. The elliptical contours of the function $\sum_{i=1}^{n}(y_i - \sum_{j}^{p} x_{ij}\beta_j)$ is shown by the solid curves in both panels; the center of the contours is the ordinary least square (OLS) solution; the constraint region of LASSO is the rotated square in the left panel, and the constraint region of ridge regression is the circle in the right panel. The LASSO (ridge) solution is the first place that the contours touch the square (circle). For LASSO, this will sometimes occur at a corner of the square, corresponding to a zero coefficient; for ridge regression, there is no corner for the contours to hit and hence zero coefficients will rarely occur. (Figure taken from [56].)

LASSO's L1 penalty will make the solutions nonlinear in the y_i, so usually one has to use a quadratic programming (QP) algorithm to compute the solution. The computation complexity of QP has prohibited LASSOs wide application in practice. This issue is alleviated after Efron et al. [104] proposed Least Angle Regression (LARS) and showed that for LASSO, the solution path in $\boldsymbol{\beta}$ space is piecewise linear and gave efficient algorithms for tracking the path. Efron et al. [104] derived the LARS algorithm which could be used to compute the LASSO solution while reducing the computational burden by at least an order of magnitude.

Similar to ridge regression, LASSO also has a Bayesian interpretation, with the sole difference to be that the prior for LASSO is the Laplace distribution

$$\boldsymbol{\beta} \sim Laplace(\mathbf{0}, \boldsymbol{b})$$

.

4.3.3 Elastic Net

Both ridge regression and LASSO are a special case of bridge regression [77],

$$\hat{\boldsymbol{\beta}}^{bridge} = argmin \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} (x_{ij}\beta_j))^2 + \lambda \sum_{j=1}^{p} |\beta_j|^q \right\} \tag{4.20}$$

When $q = 1$, the bridge regression is the same as LASSO; when $q = 2$, it becomes ridge regression.

Fan and Li [78] showed that a bridge regression penalty L_q with $1 < q < 2$ is strictly convex and has a grouping effect (qualitatively speaking, a regression method exhibits the grouping effect if the regression coefficients of a group of highly correlated variables tend to be equal (up to a change of sign if negatively correlated)[79].), but does not produce a sparse solution. This motivates Zou and Hastie [79] to use the elastic net penalty, which is a mixture of the L1 penalty from LASSO and the L2 penalty from ridge regression,

$$\hat{\boldsymbol{\beta}}^{elastic\_net} = argmin \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} (x_{ij}\beta_j)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2 \right\} \tag{4.21}$$

The elastic net is strictly convex when $\lambda_2 > 0$, which makes it have a grouping effect. This useful in the analysis of microarray data, as it tends to bring related genes into the model as a group. It also appears to give better predictions than LASSO when predictors are correlated, and in high dimensional settings. Elastic net can also produce sparse solutions due to the inclusion of L1 penalty.

A comparison of ridge regression, lasso, and elastic net is shown in Figure 4.6.

4.4 Dimension Transformation Methods

In this section, we present two dimension-reduction methods, principle component analysis (PCA) [50] and partial least squares (PLS) [50], both of which are based on

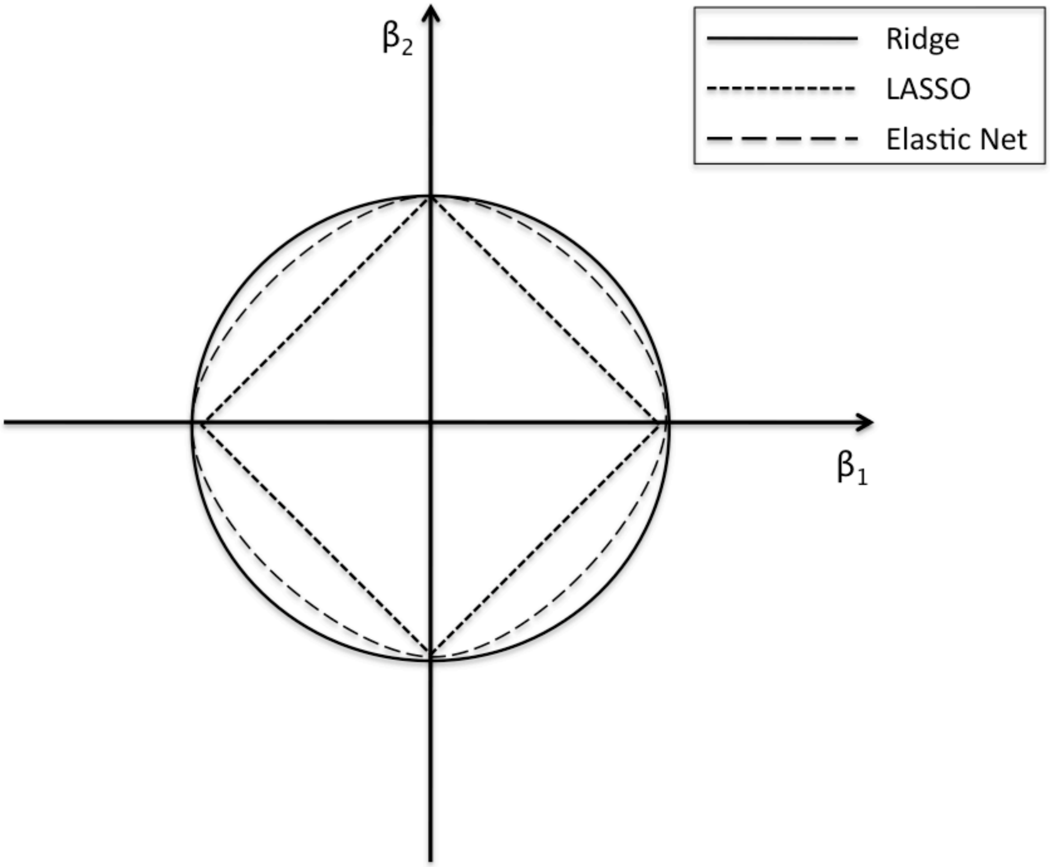


Figure 4.6: Comparison of ridge regression, LASSO, elastic net using two-dimensional contour plots. The elastic net penalty is $\alpha = \frac{\lambda_2}{\lambda_1+\lambda_2} = 0.5$; we see that singularities at the vertices and the edges of elastic net contour is strictly convex; the strength of convexity varies with α.

orthogonal linear transformation that transforms the data to a new coordinate system (dimension transformation). These two methods are very effective for dimension reduction when the input variables used in a regression are highly correlated.

4.4.1 Principal Component Analysis (PCA)

Dimension reduction of the p-dimensional space by PCA is achieved by constructing principal components (PCs), which are linear combinations of the original p predictor/explanatory variables. More precisely, in PCA, orthogonal linear combinations are constructed to maximize the variance of the linear combination of the explanatory variables sequentially,

$$\boldsymbol{w}_1 = argmax_{||\boldsymbol{w}||=1} var(\mathbf{X}\boldsymbol{w}) = argmax_{||\boldsymbol{w}||=1} \boldsymbol{w}'\mathbf{X}'\mathbf{X}\boldsymbol{w}$$

. With the first $k-1$ components, the kth component can be computed by

$$\boldsymbol{w}_k = argmax_{||\boldsymbol{w}||=1} var((\mathbf{X} - \sum_{i=1}^{k-1} \boldsymbol{w}_i \boldsymbol{w}_i^T \mathbf{X})\boldsymbol{w})$$

The principal components $\boldsymbol{s}_k = \mathbf{X}\boldsymbol{w}_k$ are subject to the orthogonality constraints $\boldsymbol{s}_k'\boldsymbol{s}_j = w_k'\mathbf{X}'\mathbf{X}w_j = 0$, for all $1 \leq j < k$. Here $\mathbf{X}$ is the $n \times p$ input matrix. The maximum number of nonzero components is the rank of $\mathbf{X}$. The kth step of PCA seeks the strongest mode of variation and the $k-1$ orthogonality constraints are imposed to ensure that the kth linear combination identifies a mode of variation distinct from those previously identified (by the previous $k-1$ components).

From geometrical perspective, PCA projects the data along the directions where the data varies the most. These directions are determined by the PCs, i.e. the eigenvectors of the covariance matrix of $\mathbf{X}$. The magnitude of the eigenvalues corresponds to the variance of the data along the eigenvector directions.

Prediction using standard methods can be carried out in the reduced space by using the constructed PCs. For instance, prediction of a continuous response vector, $\mathbf{y}$, based on the constructed PCs is the well-known principal component regression (PCR) method [80]. A PCR model is the linear regression model based on the subspace spanned by K PCs, $\{s_1, \cdots, s_K\}$:

$$y = \beta_0 + \sum_{i=1}^{K} \beta_i s_i \tag{4.22}$$

where y is the response variable, and $\{\beta_i\}_{i=0}^{K}$ are the coefficients of the new linear model.

In practice, cross-validation is used to determine the optimal number of dimension, K.

4.4.2 Partial Least Square (PLS)

PCA reduces dimension by constructing and using linear combinations that maximize the variance-based objective function, namely $var(\mathbf{X}w)$. A parallel formulation can be made for PLS, but with an objective function based on covariance. More precisely, PLS components are linear combinations of the predictor variables, constructed to maximize an objective criterion based on the sample covariance between $\mathbf{y}$ and $\mathbf{X}w$, namely $cov(\mathbf{X}w, \mathbf{y})$). Thus, the kth PLS component is obtained by finding the weight vector, w, satisfying

$$\boldsymbol{w}_k = argmax_{ww'=1} cov(\mathbf{X}w, \mathbf{y}) = argmax_{ww'=1} w'\mathbf{X}'\mathbf{y} \tag{4.23}$$

Similar to PCA, the PLS components $\boldsymbol{t}_k = \mathbf{X}\boldsymbol{w}_k$ are subject to the orthogonality constraints $\boldsymbol{t}'_k \boldsymbol{t}_j = w'_k \mathbf{X}'\mathbf{X} w_j = 0$, for all $1 \leq j < k$. The maximum number of PLS components is at most the rank of $\mathbf{X}$. Analogous to PCR, a PLS regression model with K PLS components is based on the subspace spanned by the first K PLS components, $\{t_1, \cdots, t_K\}$,

$$y = \beta_0 + \sum_{i=1}^{k} \beta_i t_i \tag{4.24}$$

where y is the response variable, and β_i is the coefficient of the new linear model.

In seeking dimension reduction useful for prediction, the objective criterion of PLS may be more sensible than PCA since there is no a priori reason why components with high predictor variation should be strongly related to the response variable, while PLS incorporates both response and predictor information into the dimension reduction process.

Chapter 5

Feature Selection Approaches for Estimating Breeding Value

In this section, we focus on the application of feature selection techniques in the QTL mapping problem. In contrast to Principal Component Analysis (PCA) and Partial Least Square (PLS), which perform dimension reduction based on projection, feature selection techniques do not alter the original representation of the variables, but merely select a subset of them. Thus, they preserve the original semantics of the variables, hence, offering the advantage of interpretability.

High dimensional data in the field of bioinformatics, which could contain high degree of irrelevant and redundant information, may cause serious problems to many machine-learning algorithms with respect to scalability and learning performance. Therefore, feature selection becomes very necessary for machine learning tasks when facing high dimensional data nowadays.

There are many objectives of feature selection, with the most important ones being: (a) to avoid overfitting and improve model performance; (b) to provide faster and more cost-effective models for further study and (c) to gain a deeper insight into the underlying processes that generated the data. However, the advantages of feature selection techniques come at a certain price, as the search for a subset of relevant features introduce an additional layer of complexity in the modeling task. Instead of just optimizing the parameters of the model for the full feature subset, we now need to find the optimal model parameters for the optimal feature subset.

We consider in-fold feature selection when applying the feature selection methods, *i.e.* feature selection will be based on only the training data instead of the whole dataset. Whole-dataset feature selection is likely to find the features that will only work well on the current dataset, but will not generalize well on future data.

In-fold feature selection only uses the training data, and the performance of the features found will be verified on the separate test data. In this way, in-fold feature selection ensures that the features that work well on both training data will similar good performance on the test data. Hence the “good” features selected by in-fold feature selection will be more likely to generalize well on future dataset.

Next, we talk four feature selection methods that are tried in our problem: Correlation-Based Feature Selection, Logic Regression, M5 Prime for linear regression and Haplotype Block.

5.1 Correlation-Based Feature Selection

The correlation-based method evaluates features individually by measuring their correlation with the response variable. The correlation is measured by the linear correlation coefficient. For a pair of vectors $\boldsymbol{x}, \boldsymbol{y}$, the linear correlation coefficient $r_{\boldsymbol{xy}}$ is given by

$$r_{\boldsymbol{xy}} = \frac{\Sigma_i (x_i - \overline{\boldsymbol{x}})(y_i - \overline{\boldsymbol{y}})}{\sqrt{\Sigma_i (x_i - \overline{\boldsymbol{x}})^2}\sqrt{\Sigma_i (y_i - \overline{\boldsymbol{y}})^2}} \tag{5.1}$$

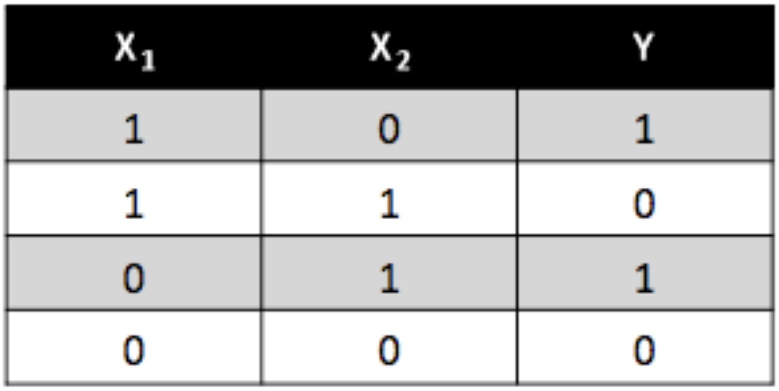
where $\overline{\boldsymbol{x}} = \frac{1}{k}\Sigma_{i=1}^{k} x_i$ is the mean of $\boldsymbol{x}$, and $\overline{\boldsymbol{y}} = \frac{1}{k}\Sigma_{i=1}^{k} y_i$ is the mean of $\boldsymbol{y}$. The value of r is between -1 and 1, inclusive. If $\boldsymbol{x}$ and $\boldsymbol{y}$ are completely (anti) correlated, r takes the value of 1 or -1; if $\boldsymbol{x}$ and $\boldsymbol{y}$ are totally independent, r is zero. So we usually take the absolute value of r. An higher value of $|r|$ means that the two vectors are more correlated.

Using the correlation-based feature selection method, the correlation between each feature and the response variable is evaluated one by one, and then all the features are ranked by the value of correlation coefficient. The top N features are selected for prediction purpose. The value of N is often determined by cross-validation.

5.2 Logic Regression

Since the correlation-based feature selection looks at one feature at a time, it is a uni-variate feature selection method. However, the uni-variate feature selection for SNP dataset can be inadequate from both statistical and biological point of view. First, the most discriminatory SNPs identified individually do not necessarily constitute the best classifier when put together [83]. Second, it is biologically inappropriate to examine SNPs in separation, given that multiple genetic markers usually function in a correlated network. A greedy searching algorithm ignoring the SNP/SNP interaction, such as the uni-variate selection, tends to include elements contributing highly redundant information. The extreme case is when two markers are exact duplicates, in which case one marker can be eleminated.

Consider two input variables X_1 and X_2, and a class variable Y. Attributes X_1 and X_2 are said to be only dependent to each other conditioned on Y. A simple example illustrating this kind of interaction is an XOR (exclusive OR) model, shown in Figure 5.1. Only considering one attribute at a time would result in conclusion that neither of the input variables correlate with Y. Looking only at X_1, for example, would result in the conclusion that Y is independent of X_1, because $p(Y = 1|X_1 = 1) = p(Y = 1|X_1 = 0) = p(Y = 1) = \frac{1}{2}$. To accurately predict the classifier variable in this interaction model, one must take both input variables into account simultaneously.



X_1	X_2	Y
1	0	1
1	1	0
0	1	1
0	0	0

Figure 5.1: An example of XOR model. Looking only at X_1 will result in the conclusion that X_1 is independent of Y (similar for X_2). However, taking both variables into consideration will lead to the correct XOR relation between X_1, X_2 and Y.

Logic Regression [76, 84] is an adaptive regression methodology mainly developed to explore high-order interactions in genomic data. Logic Regression is intended for situations where all predictors are binary (0/1) and the goal is to find Boolean combinations of these predictors that are associated with an outcome variable. As SNP data are effectively binary (as we will see later), Logic Regression is potentially useful for detecting interactions among SNPs. The objective of logic regression is partly to reduce the prediction error, but more to explore models in a novel search space that might reveal important variables and interactions, which would otherwise go unnoticed.

First, assume that all predictors X_i are binary, and use X_i^1 for $X_i = 1$ and X_i^0 for $X_i = 0$. Let Y be a phenotype trait that can be either binary or quantitative. The logic regression model is

$$\mathcal{G}(E(Y|\boldsymbol{X})) = \beta_0 + \sum_{i=1}^{k} \beta_i L_i(\boldsymbol{X}) \tag{5.2}$$

where $\mathcal{G}$ is an appropriate link function (such as the logic function), $E(Y|\boldsymbol{X})$ is the expectation of Y given X, $\boldsymbol{X}$ is a set of covariates, $\beta_0, \cdots, \beta_k$ are coefficients, and $L_1, \cdots, L_k$ are the so-called logic expressions, which are the boolean combinations of the covariates, such as $X_1^0 \wedge (X_2^1 \vee X_3^1)$. The logic expressions can be easily represented by the tree form, which is called the logic tree; see Figure 5.2.

Using the logic tree representation, it is possible to obtain any logic tree by a finite number of operations, such as growing of branches, pruning of branches and removing of leaves, etc; see Figure 5.2.

A score function could be defined for a particular configuration of the logic trees used in the model (Eq. 5.2). For linear regression, the score function could be the minus of the residue sum of squares; for classification problem, it could be the classification accuracy rate. In regular logic regression, using a simulated annealing algorithm, the logic trees in the model are selected adaptively to achieve higher scores.

We start with $L = 0$ that contains zero boolean expressions. Then, at each stage a new tree is selected at random among those that can be obtained by simple operations on the current tree. This new tree always replaces the current tree if it has a better score than the old tree, and otherwise is accepted with a probability that depends on the difference between the scores of the old and the new tree, and the stage of the algorithm. Early on, trees with considerably worse scores are still accepted, while after many iterations and toward the end of the algorithm, the probability of accepting a tree with a worse score becomes eventually almost zero. In this simulated annealing algorithm, each covariate could end up in multiple trees. Note that the dimensionality of the model (Eq. 5.2) is not the number of covariates, which may be very large, but the number of parameters, which is the number of logic trees L_k plus one, and is usually small.

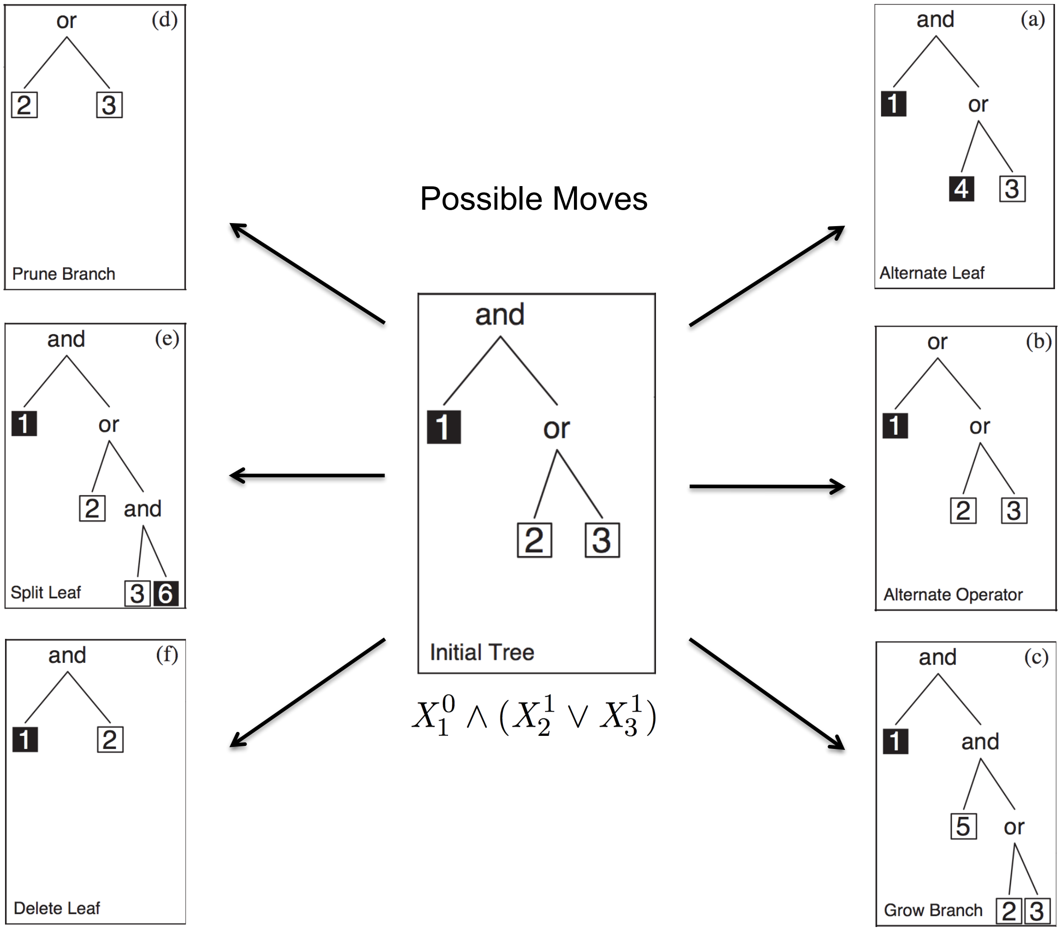


Figure 5.2: Logic tree representation of the logic expressions. The number in the box indicates the index of the covariate. The box with white background represents $X_i = 0$; and the box with black background represents $X_i = 1$. The expression $X_1^0 \wedge (X_2^1 \vee X_3^1)$ is represented by the "Initial Tree" at the center of the figure. The figures around show the possible moves allowed to grow the logic tree. (Figures taken from [76].)

As the model that best fits the training data will often overfit, cross-validation is often used to select the number of trees ($L_i(\boldsymbol{X})$) in the model and the maximum number of leaves allowed in each tree. Alternatively, a set of randomization tests can also be used to reduce the chance of overfitting [76].

In a simulated GAW12 dataset [85], logic regression successfully identified an interaction between QTL and the sequence of gene that influenced the phenotype.

Extensions to logic regression have been proposed to make it become a feature selection method. Kooperberg *et al.* [86] incorporate Monte Carlo Markov Chain (MCMC) model selection techniques to identify a group of SNPs that are potentially associated with phenotypic traits. Unlike strategies that focus on identifying a single best model that relates SNPs to the clinical outcome of interest, Monte Carlo logic regression generates a subset of SNP interactions (that is a certain part of the generated logic tree) that may be significantly associated with a trait and are selected for further investigation. Although a large number of potential logic regression models may not stand up to a rigorous 5% significance level individually, jointly they may provide strong evidence of association, which may be indicative of a genetic pathway. Monte Carlo logic regression has been applied to the study of heart disease [87].

Schwender *et al.* [89] proposed a subset-based approach in which the default search algorithm of logic regression, *i.e.* simulated annealing, is applied to different subsets of the data. More specifically, first, a bootstrap sample of the same sample size is drawn from the dataset of interest. Second, a logic regression model is constructed based on the bootstrap sample, and the logic expressions in the fitted model are converted into disjunctive normal forms (DNF) consisting of prime implicants, *i.e.* minimal AND-combinations. The DNF of the logic expression $L = (X_1^1 \wedge X_2^0) \vee (X_3^1 \wedge (X_4^0 \vee X_5^1))$ displayed in Figure 5.3 is, *e.g.*, given by

$$L = (X_1^1 \wedge X_2^0) \vee (X_3^1 \wedge X_4^0) \vee (X_3^1 \wedge X_5^1)$$

The advantage of the DNF is that interactions are directly identifiable since they are given by the AND-combinations. The above logic expression, e.g., consists of the three prime implicants $X_1^1 \wedge X_2^0$, $X_3^1 \wedge X_4^0$, $X_3^1 \wedge X_5^1$ and is true if at least one of these conjunctions is true. Schwender [88] presents a fast algorithm based on matrix algebra for generating such a DNF of a logic expression.

Finally, the process above is repeated for a given number of times, and the prime implicants that occur most frequently are selected for further analysis. In this study, we adopt the bootstrap version of Logic Regression for feature selection in the SNP dataset.

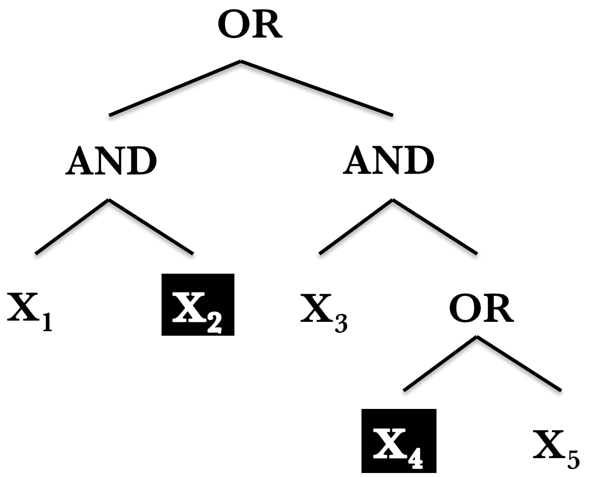


Figure 5.3: Logic tree representation of the logic expression $L = (X_1^1 \wedge X_2^0) \vee (X_3^1 \wedge (X_4^0 \vee X_5^1))$.

5.3 M5 Prime for linear regression

M5 Prime is a feature selection method for linear regression proposed in the Weka machine-learning package [90]. The basic idea behind the M5 Prime is model selection using Akaike Information Criterion (AIC). For any model M *w.r.t.* $df = D$, the AIC is given by

$$AIC(M, D) = 2|M| - 2\ln(P(D|M, \hat{\theta})) \tag{5.3}$$

where $|M|$ is the number of parameters in the statistical model, $P(D|M, \hat{\theta})$ is the maximized value of the likelihood function for the estimated model, and $\hat{\theta}$ is the maximum likelihood estimate of the model parameters θ. Lower AIC values indicate a better model. Hence AIC not only rewards goodness of fit, but also includes a penalty that is an increasing function of the number of parameters in the model. The AIC methodology attempts to find the model that best explains the data with a minimum of free parameters.

In the case of multivariate linear model, AIC is then given by

$$AIC(p, n, RSS) = 2(p+1) + n\left[\ln(\frac{RSS}{n}) + 1\right] \tag{5.4}$$

where p is the number of covariates, n is the number of observations, and RSS is the residue sum of squares given by

$$RSS = \sum_{i=1}^{n} (\boldsymbol{y}_i - f(\boldsymbol{x}_i))^2 \tag{5.5}$$

where $\boldsymbol{y}$ is the target response variable, $\boldsymbol{x}$ is the predictor variable, and $f(\boldsymbol{x})$ is the regression function in a standard regression model.

M5 prime method adopts the greedy backward elimination for feature selection by stepping through the covariates removing the one with the smallest standardized coefficient until no improvement is observed in the score given by the AIC. Like logic regression, M5 prime also consider the interactions between the covariates, but the greedy nature makes it have a tendency towards local optima.

5.4 Haplotype Block

Unlike the three methods discussed above, the haplotype block method uses the biological background knowledge to try to find the most relevant SNPs for predicting quantitative traits.

The motivation of this method is that when an individual inherits a chromosomal material from each of the parents, a recombination event can break the parental chromosomes into non-random inheritable segment, causing set of SNPs within the segment to be inherited together with high probability, and preventing random combinations of all possible SNP states within the segment. Since the recombination sites are non-uniformly distributed across the genome, recombination events in chromosomes in a population over generations lead to a block structure in SNPs on the chromosome.

As illustrated in Figure 5.4, each chromosome is a mosaic of ancestor chromosomes. Since a chromosome segment carrying the true association SNP can be inherited as a block, we can take advantage of this block structure to increase the power of the study for detecting association by considering a block of linked SNPs jointly rather than a single SNP at a time. Formally, those block structures are called *Haplotype Blocks*. We apply the idea of haplotype blocks as a feature selec-

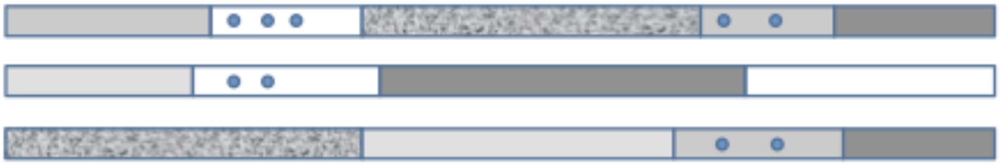


Figure 5.4: Illustration of the block structure of chromosome. The segments of SNPs of the same color have been inherited from the same ancestor chromosome. The SNPs that are of the true association are indicated as circles. [105]

tion method by only considering the SNPs that belong to a haplotype block, ignoring the rest.

Haploview [106] is comprehensive suite of tools for a wide variety of haplotype analysis. We use *Haploview* to assign the SNPs to various haplotype blocks in our study. *Haploview* implements three ways to generate haplotype blocks, “Confidence Intervals” [107], “Four Gamete Rule” [108], and “Solid Spine of Linkage Disequilibrium (LD)” [106]. We try all three in our experiment.

We apply haplotype block to do feature selection as follows. We first generate the haplotype blocks (using one of three ways) by *Haploview*. In each of the haplotype block, there is a number of SNPs, and we build a subset of the original SNP dataset that contains only the SNP features in the block. We then apply Principle Component Analysis (PCA) (see Section 4.4.1) to reduce the dimension of the subset, and generate the top Principle Components (PCs) to use in the final regression analysis; see Figure 5.5.

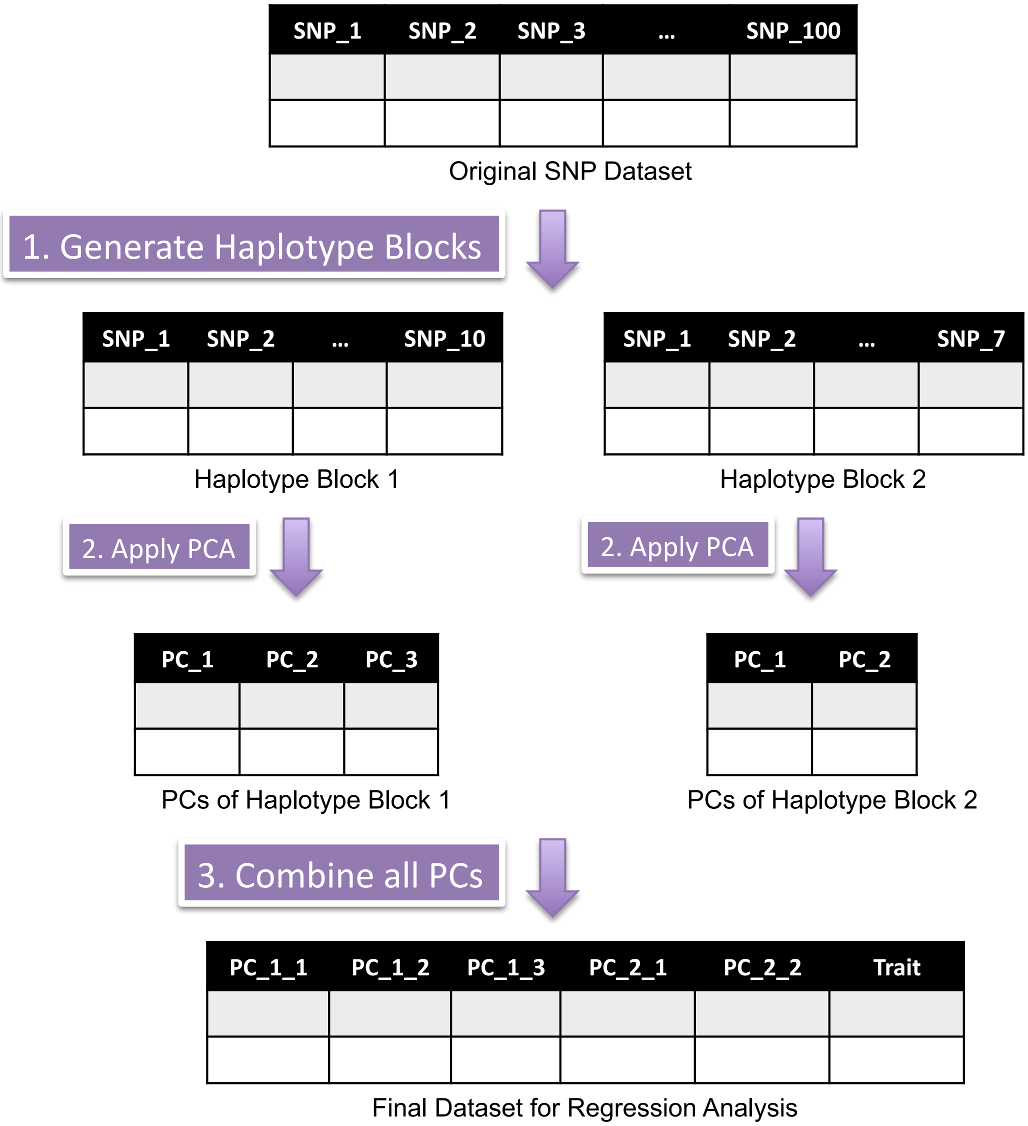


Figure 5.5: Illustration of using haplotype block for feature selection. In this example, we have an original SNP dataset that contains 100 SNPs. (Note this is just an example. The real dataset used in the study contains 1341 SNPs.) *Haploview* generate two haplotype blocks for this dataset, with one block containing 10 SNPs and the other containing 7 SNPs (The other 83 SNPs do not belong to any of the haplotype blocks.). We then apply PCA on each of the subset of the original dataset defined by the haplotype blocks. In the example, we select top 3 PCs for the first block and another top 2 PCs for the second block, then we combine the PCs and get the final dataset that contains 5 PCs for regression analysis. As we can see from the final dataset, we reduce the dimension from the original 100 SNPs to 5 PCs.

Chapter 6

Experiments and Results

In this section, we first give a brief introduction to the Bovine dataset used in this study. We then show the experiment results on the prediction accuracy of quantitative traits of the seven regression methods described in Chapter 4, along with the four feature selection methods described in Chapter 5. We finish this chapter with a discussion of the empirical results and possible future works.

6.1 Bovine Dataset Overview

The dataset used in this study comes from a diary-industry breeding program. The dataset consists of 462 dairy sires (observations). The data provider withheld 157 out of the 462 observations as the final test set for evaluation of our methods. We trained on the remaining 304 observations. 1341 SNPs are genotyped for each sire. Each SNP could only take 3 values: "1" (Homozygous Major), "2" (Heterozygous), and "3" (Homozygous Minor). We consider 5 seperate studies based on predicting 5 phenotypic traits: *FatEBV*, *FatPercentEBV*, *MilkEBV*, *ProteinPercentEBV*, and *ProteinPercentEBV*.

6.2 A First Look at the Bovine Dataset

As mentioned, a particular SNP is a categorical feature, which can only take value from "1", "2", and "3". Here, we explore how many "1"s, "2"s and "3"s are for each bull. For each bull, we have a vector of length 1341, as we collect 1341 SNPs for each bull. Then we count the numbers of "1"s, "2"s, and "3"s in the vectors for all the 304 observations. This produces three vectors of length 304, which are summerized as the three "boxplots" shown in the left diagram of Figure 6.1.

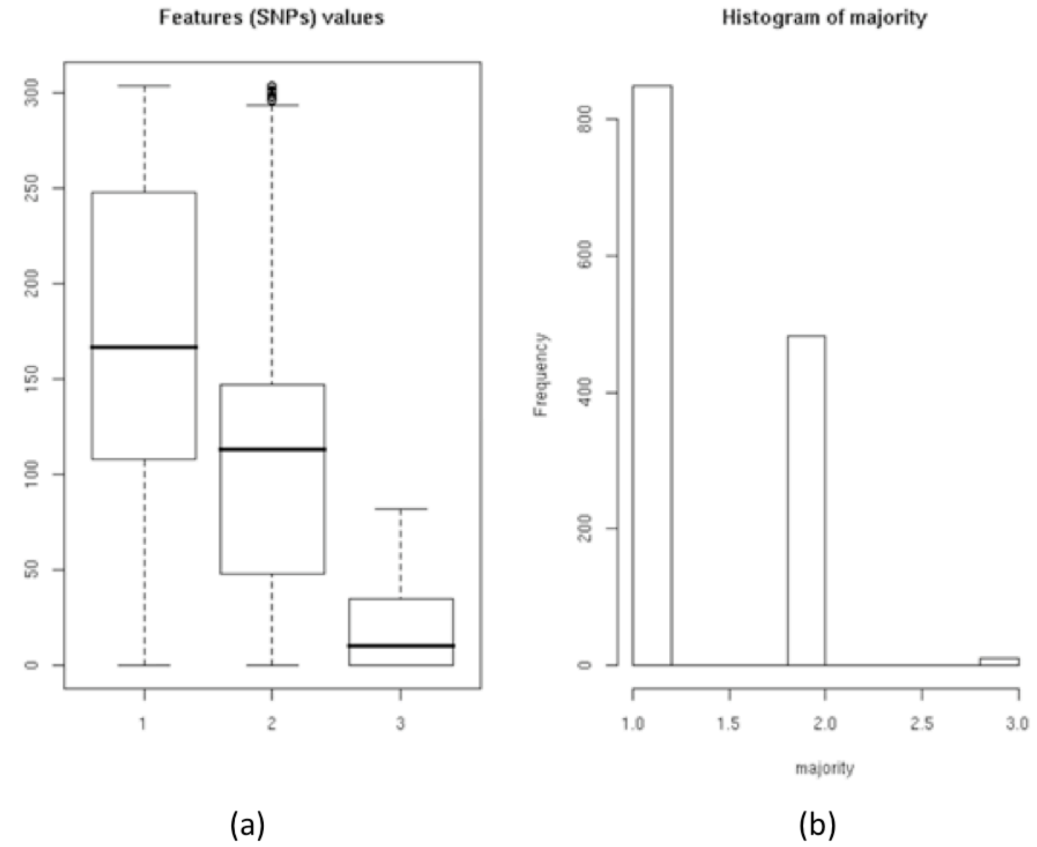


Figure 6.1: SNP (feature) facts of the Bovine SNP data. (a) is the histogram of SNP values, where the x-axis is the three possible values for each SNP: “1”, “2”, and “3”; the y-axis is the distribution of SNPs in each tuple. (b) is the histogram of the majority of SNP values, where the x-axis is the three possible values for each SNP: “1”, “2”, and “3”; y-axis is the distributions of SNPs that has that specific majority value.

In each boxplot, the bold horizontal bar in the middle shows the median value of the data. The top of the box above the median shows the 75th percentile, and the bottom of the box below the median shows the 25th percentile. Inside the box lies the middle 50% of the data. The whiskers show the maximum and minimum values of the data.

For example, the first boxplot on the left shows the median number of "1"s of the 304 observations is around 160 out of 304, and the 75th percentile is around 248, etc. From the left diagram, we can see that the occurrence of "1"s is far more frequent then that of "3"s with "2"s in-between, which matches our expectation since "1" represents "Homozygous major", while "3" represents "Homozygous minor".

In order to compare the occurrences of "1"s, "2"s, and "3"s more directly, we plot the diagram on the right of Figure 6.1. Here, the x-axis represents the "majority value" ("1", "2", and "3"), and the y-axis is number of SNPs which have that "majority value". We define the "majority value" for a particular SNP to be the value ("1", "2", or "3") that occurs most frequently. We find the majority value of all the 1341 SNPs. From the diagram we can see that around 840 out of 1341 SNPs' "majority values" are "1", while around 500 SNPs' "majority values" are "2". But to our surprise, there is one SNP whose "majority value" is "3", which should not happen. We took a close look at that SNP, and found that its values for each observation are all "3"s, which was apparently a mistake from data collecting. That SNP was removed from the dataset.

There are 5 response variables (*FatEBV*, *FatPercentEBV*, *MilkEBV*, *ProteinPercentEBV*, and *ProteinPercentEBV*), whose value we try to predict for each bull (based on its SNP profile). Figure 6.2 plots the histograms for each of the response variables. Most of these histograms appear like Gaussian distribution, in that the values have a tendency to occur more often around its mean value.

6.3 Data Pre-processing

As we mentioned earlier, one of difficulties with the SNP datasets is that the number of features (SNPs) far exceeds the number of observations. In our case, we have 1340 SNPs (after removing the all value-"3" SNP) and only 304 observations. So before feeding the dataset to the machine learning methods, we first remove the features (SNPs) that we think are not informative for prediction purpose.

The previous removed all value-"3" SNP reminds us that the SNPs whose values

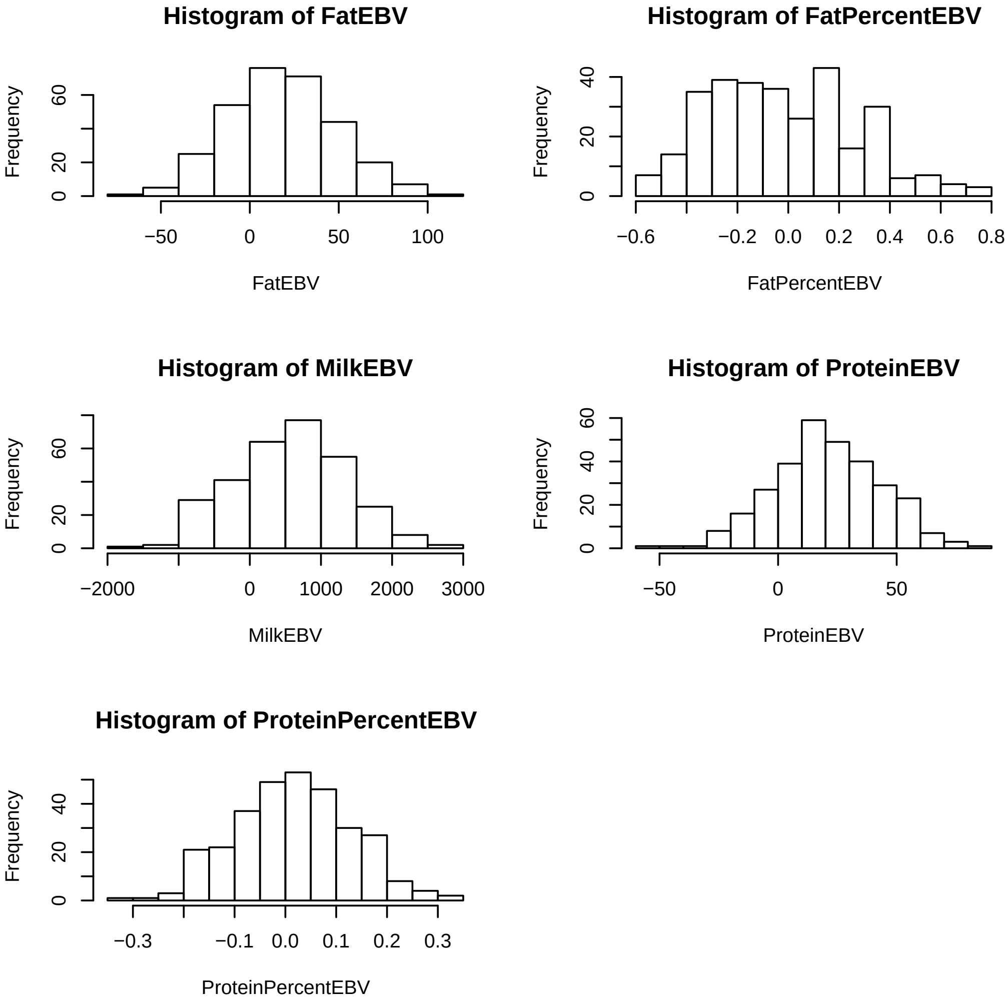


Figure 6.2: Histograms of 5 response variables of the Bovine SNP dataset, *FatEBV*, *FatPercentEBV*, *MilkEBV*, *ProteinPercentEBV*, and *ProteinPercentEBV*. The x-axis of each histogram is the range of values, while the y-axis is the frequency, *i.e.* the proportion of cases that fall into the bin.

are all "1"s and all "2"s might not be very informative as well. So our first preprocessing step is to remove all the SNPs, whose values are all the same. We found 165 such SNPs, which are more than 10% of the total 1341 SNPs. After removing these SNPs, we still have 1175 SNPs remaining.

Genetic dataset frequently contain missing values, however, most down-stream analyses require complete data. In the bovine SNP dataset, we found that almost each observation contains a number of SNPs that are recorded as missing data. In the literature many methods have been proposed to estimate missing values using information of the correlation patterns within the dataset. Each method has its own advantages, but the specific conditions for which each method is preferred remains largely unclear. Troyanskaya *et al.* compared a variety of algorithms and concluded that two methods, k-Nearest-Neighbors (KNN) and singular value decomposition (SVD), performed well in their test data sets to impute missing data in the microarray datasets [92]. Oba *et al.* [94] proposes another imputation method for missing values, which is based on Bayesian principal component analysis. For a substantial evaluation of various Missing Value (MV) imputation methods on microarray dataset, please refer to [95, 96, 97].

We tried two MV imputation methods on the bovine SNP dataset. The Majority MV method is to replace the missing value by the majority value of the corresponding feature, *i.e.* the most frequent value for that feature. For example, if 300 values out of 304 is "1" for that feature, we say the majority value of that feature is "1". Then the majority value "1" will replace each of the missing values.

The second method we tried is to use Naive Bayes classifier for MV imputation. Naive Bayes is one of the most effective and efficient classification algorithms. Assume that $A_1, \cdots, A_n$ are n attributes. An instance I is represented by a vector $(a_1, \cdots, a_n)$, where a_i is the value of A_i. Let C represent the class variable and c represent the value that C takes. In general, a Naive Bayes classifier is defined as follows.

$$\mathcal{G}(a_1, \cdots, a_n) = argmax_c\{p(c|a_1, \cdots, a_n)\} = argmax_c\{p(c) \cdot p(a_1, \cdots, a_n|c)\} \quad (6.1)$$

where $p(c)$ is the marginal probability of class c, and $p(a_1, \cdots, a_n|c)$ is the conditional probability of $A_1 = a_1, \cdots, A_n = a_n$ given $C = c$. In Naive Bayes, all attributes are assumed independent given the class; that is

$$p(a_1, \cdots, a_n|c) = \prod_{i=1}^{n} p(a_i|c) \tag{6.2}$$

Therefore, Eq. 6.1 can be written as,

$$\mathcal{G}(a_1, \cdots, a_n) = argmax_c \left\{ p(c) \cdot \prod_{i=1}^{n} p(a_i|c) \right\} \tag{6.3}$$

Figure 6.3 shows the structure of Naive Bayes classifier, where each attribute node has the class node as its parent, but does not have any parent from attribute nodes. Because the values of $p(a_i|c)$ can be easily estimated from training instances, Naive Bayes is easy to construct.

Despite the fact that the unrealistic independence assumptions are often inaccurate, the naive Bayes classifier has several properties that make it surprisingly useful in practice [98]. In particular, the decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one dimensional distribution. This in turn helps to alleviate problems stemming from the curse of dimensionality, such as the need for data sets that scale exponentially with the number of features. Like all probabilistic classifiers under the MAP decision rule, it arrives at the correct classification as long as the correct class is more probable than any other class; hence class probabilities do not have to be estimated very well. In other words, the overall classifier is robust enough to ignore serious deficiencies in its underlying naive probability model.

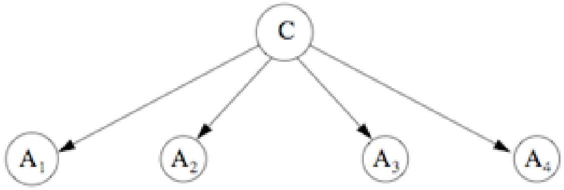


Figure 6.3: Naive Bayes classifier. The node denoted “C” is the class variable, and all the other nodes represent the attributes.

The Naive Bayes classifier is used for MV imputation as follows. For each feature (SNP) that contains missing values, we build a Naive Bayes classifier, with that feature as the class variable and all other features as attributes. We can use the observations of which that feature (the current class variable) is not missing as the training data to build the classifier, and then use it to impute the values of that feature for the rest of the observations. For example, if 1000 out of 1341 features contain missing values, we will build 1000 Naive Bayes classifiers for MV imputation.

The Majority MV method is actually a quite simple method, which motivates us to try the second method to see if a more sophiscated method will have better results. However, as we will see later, their experiment performances are quite similar. We just use the replace-by-majority method for MV imputation in most of our experiments.

After handling the missing data, we start removing uninformative features once again. But this time we remove the features whose values are 95% or more the same among the observations. We found 135 such features, and leaves us with $1175 - 135 = 1040$ features.

This dataset with the 1040 features is regarded as the "original dataset". We also consider the binary representation of the original dataset, where each feature in the original dataset is represented by two binary features in the "binary dataset". This can be done easily as follows,

Original SNP Value	Binary Representation
1	0, 0
2	1, 0
3	1, 1

Table 6.1: Transform the "original dataset" to "binary dataset". Each feature in the original dataset is represented by two binary features in the binary dataset.

Thus, in the binary dataset, we have 2080 binary features. Again, we remove the features whose values are 95% or more the same among the observations. That produces the 1330-feature binary dataset. Our empirical studies consider both the 1040-feature original dataset and the 1330-feature binary dataset.

6.4 Experiment Design

As we mentioned earlier, we tried two approaches for our problem, one with feature selection, and the other without feature selection. In this section, we introduce the experiment procedure for the two approaches.

Figure 6.4 shows the main experimental procedure. The first step is data preprocessing, which was mentioned in the previous section. Secondly, we divide the 304 observations into 10 folds, where 9 folds are used as training data and the remaining

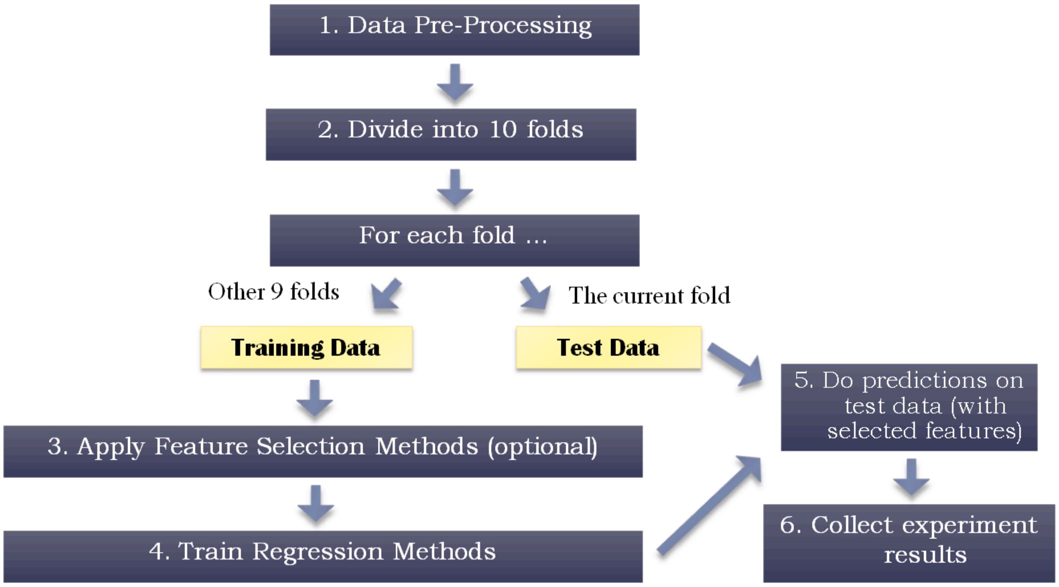

1. Data Pre-Processing
2. Divide into 10 folds
For each fold ...
Other 9 folds
The current fold
Training Data
Test Data
3. Apply Feature Selection Methods (optional)
4. Train Regression Methods
5. Do predictions on test data (with selected features)
6. Collect experiment results

Figure 6.4: Experimental procedure used to evaluate the performance of the methods.

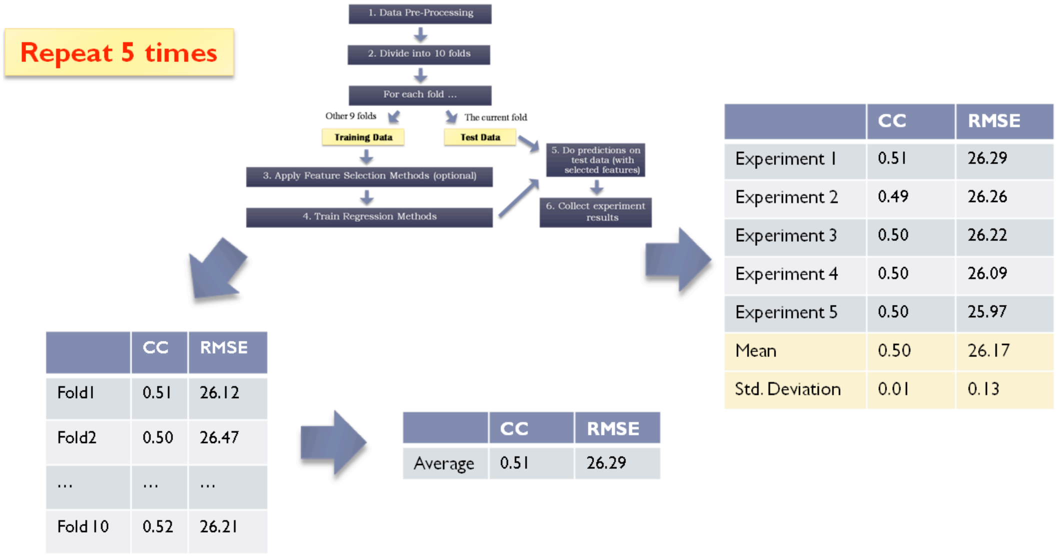
1 fold as test data. The third step is the optional in-fold feature selection, i.e. feature selection will be based only on the training data. Fourthly, a regression method will be trained on the training data, which is then used to make predictions on the test data. Finally, the experiment results of the test data are collected.

The experiment steps above are repeated 10 times with a different fold used as the test data each time. The overall experiment procedure is repeated 5 times, where the 304 observations are divided differently into 10 folds. The results recorded will be the average of the 50 experiments for each combination of feature selection and regression methods. Figure 6.5 shows the overall experiment procedure.

6.5 Performance Measures

We use two performance measures to compare different combination of feature selection and regression methods: Correlation Coefficient (CC) and Root Mean Square Error (RMSE).

Correlation Coefficient (CC): CC is a measure of how well trends in the predicted values follow trends in past actual values [105]; see Eq. 5.1. It measures how well the predicted values from a forecast model “fit” with the held-out data. The range of CC is $[-1, 1]$. Both CC equals -1 and CC equals 1 indicates strong correlation, while that CC equals 0 means no correlation. We therefore redefined CC as the


Repeat 5 times
1. Data Pre-Processing
2. Divide into 10 folds
For each fold ...
Other 9 folds
The current fold
Training Data
Test Data
3. Apply Feature Selection Methods (optional)
4. Train Regression Methods
5. Do predictions on test data (with selected features)
6. Collect experiment results

	CC	RMSE
Fold1	0.51	26.12
Fold2	0.50	26.47
...	...	...
Fold 10	0.52	26.21

	CC	RMSE
Average	0.51	26.29

	CC	RMSE
Experiment 1	0.51	26.29
Experiment 2	0.49	26.26
Experiment 3	0.50	26.22
Experiment 4	0.50	26.09
Experiment 5	0.50	25.97
Mean	0.50	26.17
Std. Deviation	0.01	0.13

Figure 6.5: Overall experiment procedure, which shows that the experiment procedure shown in Figure 6.4 is repeated for 5 times, each time with a different division of 10 folds of observations.

absolute value of CC. A perfect fit gives a CC of 1.0. The closer CC is to 1, the better our prediction is.

Root Mean Square Error (RMSE): RMSE is another frequently used measure of the differences between values predicted by a forecast model $f(\boldsymbol{x})$ and the values y actually observed from the thing being modeled; see Eq. 6.4. One of the advantages of RMSE is that it has the same units as the quantity being estimated, so you will have a more direct feeling about how good the prediction is.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(f(\boldsymbol{x}_i) - y_i)^2}{n}} \tag{6.4}$$

6.6 Methods Availability

As discussed previously, we considered various combination of seven regression methods along with four feature selection methods.

The seven regression methods are

- Support Vector Machine for Regression (SVR) (RBF kernel)
- Gaussian Process (GP) (RBF kernel)
- Ridge Regression

- LASSO
- Elastic Net
- Principal Component Analysis Regression (PCA)
- Partial Least Square Regression (PLS)

Also, the four feature selection methods are

- Correlation-based feature selection
- Logic Regression
- M5 prime for linear regression
- Haplotype Block

All experiments are implemented in *R* by using the *R* packages of the regression and feature selection methods, except M5 prime (implemented in *Weka*) and Haplotype Block (using *Haploview*). The detail information of the *R* implementation for each method is as follows; (see Table 6.2)

	R Package	Version	*R* Method	Parameter selected by *CV*
SVR (RBF kernel)	kernlab	0.9-5	gausspr	C
GP (RBF kernel)	kernlab	0.9-5	kSVR	
Ridge Regression	MASS	7.2-40	lm.ridge	λ
LASSO	lars	0.9-7	lars	λ
Elastic Net	elasticnet	1.0-3	cv.enet	λ_2
PCA	pls	2.1-0	pcr	Number of principal components
PLS	pls	2.1-0	plsr	Number of PLS components
Logic Regression	logicFS	1.9-5	logreg	Number of trees, number of leaves

Table 6.2: *R* implementation details for each method, including the R package name, the method name and also the parameters of the methods whose values need to be selected by in-fold cross-validation based on the training data.

6.7 Experiment Results

First, we compare the regression methods in predicting the *FatEBV* trait based on binary dataset without feature selection; see Table 6.3. For each regression method, the results in the tables represent the statistical summary (mean and standard deviation) of the experiments. The methods tried are listed in the first column. The number before "±" is the mean of 5 repetition of the 10-fold cross-validation results, and the number after "±" is the standard deviation.

One general remark is that machine-learning kernel methods generally perform better than the other statistical methods, with GP being the best method in this experiment with an average correlation on the test data of about 0.53. But the difference between GP and SVR is minimal. Overall, machine learning methods achieve better results than the statistical methods, see Section 6.7.1 for significance tests on comparing different methods.

Method	Correlation Coefficient (CC)	$RMSE$
SVR	0.52 ± 0.01	25.46 ± 0.11
Gaussian Process	**0.53 ± 0.01**	**25.38 ± 0.11**
Ridge Regression	0.46 ± 0.03	27.48 ± 0.32
Lasso	0.48 ± 0.02	29.24 ± 0.22
Elastic Net ($\lambda_1 = 0.2$)	0.47 ± 0.02	29.04 ± 0.11
Elastic Net ($\lambda_1 = 0.4$)	0.47 ± 0.02	29.29 ± 0.11
Elastic Net ($\lambda_1 = 0.6$)	0.46 ± 0.02	29.28 ± 0.11
Elastic Net ($\lambda_1 = 0.8$)	0.48 ± 0.02	29.24 ± 0.12
PCA	0.39 ± 0.01	27.46 ± 0.07
PLS	0.47 ± 0.01	26.75 ± 0.13

Table 6.3: Experiment results on *FatEBV* without applying feature selection methods using on original dataset.

In the second experiment, we apply the two machine learning methods, GP and SVR, to predict all the 5 traits using binary dataset; see Table 6.4. The results of two methods are still quite close. Specifically, GP performs best for predicting *FatEBV*, *FatPercentEBV*, and *ProteinEBV*; SVR performs the best for the other two. By averaging the results of the 5 traits, GP performs slightly better than SVR.

	SVR	GP
FatEBV	0.51 ± 0.01	**0.52 ± 0.01**
FatPercentEBV	0.40 ± 0.01	**0.41 ± 0.01**
MilkEBV	**0.52 ± 0.01**	0.47 ± 0.01
ProteinEBV	**0.48 ± 0.01**	0.46 ± 0.01
ProteinPercentEBV	0.43 ± 0.01	**0.46 ± 0.01**

Table 6.4: SVR and GP results on all the 5 traits for prediction. (Results are measured by Correlation Coefficient.)

Next, we compare the difference between the results based on original dataset and that based on binary dataset; see Table 6.5. The results suggest that the general performance of the methods using binary data is better than that using the original data, although for some methods, like SVR and GP, the difference is very small.

Method	Binary Data	Normal Data
SVR	**0.52 ± 0.01**	0.52 ± 0.01
Gaussian Process	**0.53 ± 0.01**	0.52 ± 0.01
Ridge Regression	**0.43 ± 0.03**	0.29 ± 0.04
LASSO	**0.48 ± 0.02**	0.43 ± 0.02
Elastic Net ($\lambda_1 = 0.2$)	**0.47 ± 0.02**	0.41 ± 0.02
Elastic Net ($\lambda_1 = 0.4$)	**0.47 ± 0.02**	0.44 ± 0.02
Elastic Net ($\lambda_1 = 0.6$)	**0.46 ± 0.02**	0.44 ± 0.02
Elastic Net ($\lambda_1 = 0.8$)	**0.48 ± 0.02**	0.43 ± 0.02
PCA	**0.39 ± 0.01**	0.38 ± 0.01
PLS	**0.47 ± 0.01**	0.43 ± 0.01

Table 6.5: Comparison of experiment results between the binary representation and the original representation of the bovine SNP dataset.

Finally, we show the empirical results of the approach with feature selection (using binary dataset); see Table 6.6. We tried various combinations of feature selection and regression methods. The best combinations are logic regression with SVR/GP/PLS, whose correlation on the test data are all about 0.47, which is smaller than 0.53, the result of GP alone without feature selection. To our surprise, the performance of the methods with feature selection is worse than that without feature selection.

Table 6.7 shows more empirical results using haplotype block as the feature selection method, which are based on 3 different ways to generate haplotype blocks, “Confidence Intervals” [107], “Four Gamete Rule” [108], and “Solid Spine of Linkage Disequilibrium (LD)” [106]. We had thought that the biological background knowledge would help the feature selection process, however, to our surprise, the results using haplotype block are much worse than those using the other three feature selection methods. We guess the reason might be that the algorithms that try to find haplotype blocks only take the SNPs as input, and do not care about the quantitative traits for prediction. It might be the case that the haplotype blocks found are not significantly associated with the quantitative traits, and instead, the SNPs excluded from the blocks are actually correlated with the traits. In order

Regression Method	Feature Selection Method	Correlation Coefficient (CC)	$RMSE$
Linear Regression	M5 Prime	0.44 ± 0.01	26.57 ± 0.33
Linear Regression	Correlation-based	0.43 ± 0.01	27.81 ± 0.38
PLS	Logic Regression	0.47 ± 0.01	27.33 ± 0.23
LASSO	Haplotype Block	0.25 ± 0.03	53.28 ± 3.83
SVR	Logic Regression	0.47 ± 0.01	26.73 ± 0.28
SVR	Correlation-based	0.43 ± 0.02	28.55 ± 0.41
Gaussian Process	Logic Regression	0.47 ± 0.01	26.81 ± 0.28
Gaussian Process	Correlation-based	0.42 ± 0.02	29.23 ± 0.40
Gaussian Process	-	**0.53 ± 0.0**1	25.12 ± 0.39

Table 6.6: Experiment results of the regression methods with feature selection on the binary bovine SNP dataset. (This result using Haplotype Block as the feature selection method is using on “Solid Spine of LD” to generate the haplotype blocks. This is the best result using haplotype block as the feature selection method. More results appear in Table 6.7.)

to verify this possibility, we tried the other way around, *i.e.* using only the SNPs outside the blocks, and found that the results were actually better than that using the SNPs in the blocks (see Table 6.8), which proved that using haplotype blocks for feature selection did not work in our problem.

	Confidence Intervals	Four Gamete Rule	Solid Spine of LD
Linear Regression	0.17 ± 0.03	0.11 ± 0.04	0.20 ± 0.02
LASSO	0.21 ± 0.04	0.14 ± 0.03	**0.25 ± 0.03**
SVR	0.16 ± 0.03	0.15 ± 0.04	0.19 ± 0.03

Table 6.7: Experiment results on *FatEBV* using Haplotype Block for feature selection measured by Correlation. The first row lists three ways to generate haplotype blocks. The first column lists three regression methods tried, Linear Regression, LASSO, and SVR.

From the experiments above, there are two general findings. First, the two kernel methods, GP and SVR, are among the best methods tried in this study. Second, feature selection methods not only fail to increase the prediction accuracy, but they actually reduce it.

We recommend GP and SVR without feature selection to the data provider. Table 6.9 shows the results of both method using 304 observations as training data and the withheld 167 observations as test data. The average correlation of the 5 traits reaches 0.56 for both methods, which is much better than 0.47 (GP) and 0.46 (SVR), the average correlation from the cross-validation results using 304 ob-

	Confidence Intervals	Four Gamete Rule	Solid Spine of LD
In-block-SNPs	0.17 ± 0.03	0.11 ± 0.04	0.20 ± 0.02
Out-block-SNPs	**0.43 ± 0.02**	**0.44 ± 0.04**	**0.37 ± 0.03**

Table 6.8: Experiment results on comparing*FatEBV* using in-block-SNPs versus out-block-SNPs by applying Haplotype Block for feature selection. The first row lists three ways to generate haplotype blocks. We use Linear Regression as the regression method in this experiment.

	CC (GP)	*RMSE* (GP)	*CC* (SVR)	*RMSE* (SVR)
FatEBV	0.58	24.62	0.57	24.66
FatPercentEBV	0.60	0.24	0.60	0.24
MilkEBV	0.55	667.22	0.55	668.77
ProteinEBV	0.55	19.53	0.56	19.53
ProteinPercentEBV	0.54	0.11	0.54	0.11

Table 6.9: Experiment results on GP and SVR using 304 observations as training data and the withheld 167 observations as test data.

servations. In particular, the correlation for *FatPercentEBV* is increased by 43%, from 0.42 to 0.60. This indicates that with more observations, the prediction of quantitative traits using SNP data will be more accurate.

6.7.1 Statistic Significance Test

In order to verify that the kernel methods do perform better than the statistical methods, we apply student's t-test to compare the mean prediction accuracy of the methods. The null hypothesis is that the mean prediction accuracy of method A is the same as that of method B. Here we choose the significance level at $\alpha = 0.05$, *i.e.* 95% confidence interval. In order to reject the null hypothesis, we need a p-value to be smaller than 0.05.

As we have mentioned, for each method, we perform 50 experiments (5 repetition of the 10-fold cross-validation experiment), so the sample size for the significance test is 50. In all the experiments, we use the binary representation of the dataset and prediction target is *FatEBV*. For the Elastic Net method, we only include the one with $\lambda_1 = 0.8$, as it is the Elastic Net parameter setting that performs best in the experiment.

Table 6.10 summaries the results of the significance test. The only two p-values that are greater than 0.05 are the comparison between GP and SVR, and the com-

	SVR	GP
Ridge Regression	$1.86e^{-3}$	$8.85e^{-5}$
LASSO	$2.58e^{-3}$	$1.40e^{-4}$
Elastic Net ($\lambda_1 = 0.8$)	**0.16**	$1.15e^{-2}$
PCA	$2.34e^{-10}$	$1.14e^{-13}$
PLS	$1.46e^{-2}$	$3.28e^{-4}$
GP	**0.40**	—

Table 6.10: Statistical significance test on comparing the methods' prediction accuracy. The value in row i and column j is the p-value resulting from a student t-test comparing the method listed in the first column of row i and the method listed in the first row in column j. As we are using the 0.05 significance level, a p-value smaller than 0.05 will reject the null hypothesis, which indicates the prediction accuracies of the methods compared is really different.

parison between SVR and Elastic Net ($\lambda_1 = 0.8$). The rest results are all statistical significant at the 95% confidence interval, which verifies that the two machine-learning kernel methods do generally perform better than the statistical methods. Also, there is no obvious performance gap between GP and SVR.

6.8 Discussion

How good are our results? The result of QTL mapping has clear implications on the animal breeding industry. Additionally, the advent of SNP datasets introduces a wealth of genetic variation information for identifying the QTL associated with economically important traits. An automated or generic approach for accurate prediction of those traits and locating relevant QTL based upon the SNPs information will have a strong impact on selecting breeding animal.

In the literature of QTL mapping, some researchers have already proposed to use SNP dataset to predict complex traits, and reported very good results, but some of them were a little over-optimistic. Meuwissen *et al.* [99] reported that the correlation of GEBV with true breeding values was $0.78 - 0.85$ using Bayesian methods. However, their results are based on a simulation dataset with strict assumptions, such as equally spaced QTL always centered between two markers, and other assumptions, which may not be possible or valid, such as assuming commercial populations, being in a mutation-drift equilibrium (MDE) and a trait with heritability, $h2 = 0.5$. De Roos *et al.* [100] tried Meuwissens methods [99] on a real dairy cattle dataset with 32 markers and 1135 progeny-tested bulls that were sired by 27 grandsires, which are the “grandfather” of the bulls. One of the markers is known to have a large effect on fat percentage. They reported that the correlation reached 0.746 using the similar Bayesian methods. But when computing polygenic effects, haplotype effects, and gene effects in the multi-QTL model, the information from test data is also used, which compromises their results. Long *et al.* [101] also tried to use SNPs to predict quantitative traits of broiler sires. They first discretized the quantitative traits into binary classes, and then developed a two-step feature selection method to find the most relevant SNPs for the binary traits. They claimed that the two-step method improved classification accuracy over the case without feature selection from around 50% to above 90%. One fatal problem with their approach is that all the samples are used in the feature selection methods, which will cause their model overfit their current samples and makes their results unrealistic.

In our study, the average prediction accuracy of the 5 traits is about 0.56 using the kernel methods, GP and SVR. Although these results are not particularly good, to our knowledge, it is the first time that such a high dimensional real SNP dataset (1341 SNPs) has been used for breeding value estimation and QTL mapping. It is a very encouraging starting point, as with the availability of more bovine samples and the development of other kernel methods, we see the possibility of more accurate prediction of quantitative traits based on the genetic markers information.

Why feature selection failed in our study? One of a major objective of this study is to find the SNP markers associated with the phenotypic traits, so that the QTL of those traits could possibly be located. However, the feature selection methods tried all failed in this case.

We looked into the problem to figure out why those methods failed. We found that the features selected based on the training data could achieve nearly perfect accuracies back on the training data, but they just did not generalize well on the test data. Another finding was that a different cut of training and test data would often lead to a different subset of features to be selected based on the training data. We guess that there could be several reasons for this to happen. First, the SNP dataset might contain too much noise, i.e. quite a few SNPs appear to be highly correlated with the traits by chance, due to the relatively small sample size as compared with the number of SNPs. (That keeps the feature selection methods from discovering the true interactions.) Secondly, perhaps none of 1341 SNPs are actually very closely associated with the traits. It is suggested that more than 30,000 SNPs are needed to cover all the possible locations of QTL on the bovine genome [6]. Thirdly, perhaps the feature selection methods we tried are not powerful enough to detect the most relevant SNPs.

6.9 Future Work

For future research, we would like to verify the reasons why feature selection fail to produce any improvement in this problem. This requires exploring other feature selection methods - ones that have proved to work well for high dimensional data, like Multifactor dimensionality reduction (MDR). Also, we would like to explore ways to use some biological background information to help filter the SNPs for further analysis. Our the experiment results suggest that the binary dataset seems to be a better representation of the SNP dataset. We would like to see if there are more suitable representations, *e.g.*, using three bits to represent the values of the SNPs, instead of two.

Chapter 7

Conclusion

In this dissertation, we applied two machine-learning kernel methods, Support Vector Machine and Gaussian Process, along with five statistical regression methods, to the the SNP dataset to learn a regressor for estimating breeding value and mapping QTL. The empirical results from this study indicate that the two kernel methods could achieve better prediction accuracy than the statistical methods. We also tried several feature selection techniques in an attempt to reduce the high dimensionality of the SNP dataset and to find the most relevant SNPs associated with the traits for prediction. However, we found that these feature selection methods actually degraded prediction accuracy.

Bibliography

[1] Allard, R.W. 1960. Principles of plant breeding. New York: John Wiley.

[2] Brian K, Julius W. Identifying and incorporating genetic markers and major genes in animal breeding programs. QTL Course: June 2000 Belo Horizonte Brasil.

[3] Gupta, P.K., Roy, J.K., Prasad, M., 2001, Single nucleotide polymorphisms: a new paradigm for molecular marker technology and DNA polymorphism detection with emphasis on their use in plants. Curr. Sci. 80:524535.

[4] M. West, Bayesian factor regression models in the “large p, small n” paradigm, in: J.M. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, M. West (Eds.), Bayesian Statistics, Vol. 7, Oxford University Press, Oxford, 2003, pp. 723732.

[5] Furey, T.S. et al. Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics 2000; 16, 906914.

[6] Ben H. “QTL Mapping, MAS, and Genomic Selection” course notes, Presentation 5, 2008.

[7] Knapp, S. J., 1998 Marker-assisted selection as a strategy for increasing the probability of selecting superior genotypes. Crop Sci. 38:1164-1174

[8] Moore JH. The ubiquitous nature of epistasis in determining susceptibility to common human diseases. Hum Hered 2003, 56:73-82. 31.

[9] Gauderman WJ, Faucett CL. Detection of gene-environment interactions in joint segregation and linkage analysis. Am J Hum Genet 1997 Nov; 61 (5): 1189-99.

[10] Coffey CS, Hebert PR, Krumholz HM, et al. Reporting of model validation procedures in human studies of genetic interactions. Nutrition 2004; 20 (1): 69-73.

[11] Coffey CS, Hebert PR, Ritchie MD, et al. An application of conditional logistic regression and multifactor dimensionality reduction for detecting gene-gene interactions on risk of myocardial infarction: the importance of model validation. BMC Bioinformatics 2004; 5: 49

[12] Mitchell T. Machine learning. Boston (MA): McGraw Hill, 1997

[13] Cai YD, Doig AJ: Prediction of Saccharomyces cerevisiae protein functional class from functional domain composition. Bioinformatics 2004, 20:1292-1300.

[14] Li T, Zhang C, Ogihara M: A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. Bioinformatics 2004, 20:2429-2437.

[15] Cai YD, Liu XJ, Li YX, Xu XB, Chou KC: Prediction of beta-turnswith learning machines. Peptides 2003, 24:665-669.

[16] Dobrokhotov PB, Goutte C, Veuthey AL, Gaussier E: A probabilistic information retrieval approach to medical annotation in SWISS-PROT. Stud Health Technol Inform 2003, 95:421-426.

[17] Zhang LV, Wong SL, King OD, Roth FP: Predicting co-complexedprotein pairs using genomic and proteomic data integration.BMC Bioinformatics 2004, 5:38.

[18] Han LY, Cai CZ, Lo SL, Chung MC, Chen YZ: Prediction of RNA-binding proteins from primary sequence by a support vectormachine approach.RNA 2004, 10:355-368.

[19] Frank E, Hall M, Trigg L, Holmes G, Witten IH: Data mining in bioinformatics using Weka. Bioinformatics 2004.

[20] Quinlan JR: C4.5: programs for machine learning San Francisco, CA, USA, Morgan Kaufmann Publishers Inc; 1993.

[21] Pavlidis P, Wapinski I, Noble WS: Support vector machine classification on the web. Bioinformatics 2004, 20:586-587.

[22] Lander, ES; Botstein, D. Mapping Mendelian factors underlying quantitative traits using RFLP linkage maps. Genetics. 1989;121:185199.

[23] Ashwell M.S., Van Tassell C.P., Detection of putative loci affecting milk, health, and type traits in a US Holstein population using 70 microsatellite markers in a genome scan, J. Dairy Sci. 82 (1999) 24972502.

[24] Casas E., Stone R.T., Keele J.W., Shackelford S.D., Kappes S.M., Koohmaraie M., A comprehensive search for quantitative trait loci affecting growth and carcass composition of cattle segregating alternative forms of the myostatin gene, J. Anim. Sci. 79 (2001) 854860.

[25] Zeng, Z-B. Theoretical basis for separation of multiple linked gene effects in mapping quantitative trait loci. Proceedings of the National Academy of Sciences of the United States of America. 1993;90(23):1097210976.

[26] Jansen, R. C., 1993 Interval mapping of multiple quantitative trait loci. Genetics

[27] Cassady J.P., Johnson R.K., Pomp D., Rohrer G.A., Van Vleck L.D., Spiegel E.K., Gilson K.M., Identication of quantitative trait loci affecting reproduction in pigs, J. Anim. Sci. 79 (2001) 623633.

[28] Ikonen T., Bovenhuis H., Ojala M., Ruottinen O., Georges M., Associations between casein haplotypes and rst lactation milk production traits in Finnish Ayrshire cows, J. Dairy Sci. 84 (2001) 507514.

[29] Nguyen DV, Rocke DM. Tumor classification by partial least squares usingmicroarray gene expression data. Bioinformatics 2002; 18:3950.

[30] Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68 (1) 4967.

[31] Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. Journal of Royal Statistical Society, Series B, 67(2), 301320.

[32] Ritchie MD, White BC, Parker JS, Hahn LW, Moore JH: Optimization of neural network architecture using genetic programming improves detection and model-

ing of gene-gene interactions in studies of human diseases. BMC Bioinformatics 2003, 4:28.

[33] Ritchie, M. D., Hahn, L. W. and Moore, J. H. Power of multifactor dimensionality reduction for detecting gene-gene interactions in the presence of genotyping error, missing data, phenocopy, and genetic heterogeneity. Genet Epidemiol. 2003, 24 150157.

[34] Moore, J. H. (2007). Genome-wide analysis of epistasis using multifactor dimensionality reduction: feature selection and construction in the domain of human genetics. In: Zhu, Davidson (eds.) Knowledge Discovery and Data Mining: Challenges and Realities with Real World Data, IGI, (in press).

[35] Daz-Uriarte R, Alvarez de Andrs S: Gene Selection and Classification of Microarray Data Using Random Forest. BMC Bioinformatics 2006, 7:3.

[36] Statnikov A, Aliferis C, Tsamardinos I, Hardin D, Levy S: A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis.Bioinformatics 2005, 21(5):631-643.

[37] Cho S, Won H: Machine learning in DNA microarray analysis for cancer classification. Proceedings of the First Asia-Pacific bioinformatics Conference 2003.

[38] Dudoit S, Fridlyand J, Speed TP. Comparison of discrimination methods for the classication of tumors using gene expression data. Technical Report 576. Department of Statistics, University of California, Berkeley, CA; 2000.

[39] Li L, Weinberg CR, Darden TA, Pedersen LG. Gene selection for sample Classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. Bioinformatics 2001;17(12):113142.

[40] Furey TS, Cristianini N, Duffy N, Bednarski DW, Schummer M, Haussler andD. validation of cancer tissue samples using microarray expression data. Bioinformatics 2000;16(10): 90614.

[41] Friedman, N., Linial, M., Nachman, I. and Peer, D. Using Bayesian networks to analyze expression data. Journal of Computational Biology, 2000, 7:601-620.

[42] Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., GaasenBeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Blomfield, C. D.,

and Lander, E. S. Molecular classification of cancer: Class discovery and class prediction by gene-expression monitoring. Science, 1999, 286:531-537.

[43] Khan, J., Wei, J. S., Ringner, M., Saal, L. H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C. R., Peterson, C. And Meltzer, P. S. (2001): Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. Nature Medicine, 7(6):673-679.

[44] Bulmer, M. G. 1980. The Mathematical Theory of Quantitative Genetics. Clarendon Press, Oxford.

[45] Wong, G., B. Liu, J. Wang, Y. Zhang, X. Yang, Z. Zhang, Q. Meng, J. Zhou, D. Li, J. Zhang, P. Ni, S. Li, L. Ran, H. Li, R. Li, H. Zheng, W. Lin, G. Li, X. Wang, W. Zhao, J. Li, C. Ye, M. Dai, J. Ruan, Y. Zhou, Y. Li, X. He, X. Huang, W. Tong, J. Chen, J. Ye, C. Chen, N. Wei, L. Dong, F. Lan, Y. Sun, Z. Yang, Y. Yu, Y. Huang, D. He, Y. Xi, D. Wei, Q. Qi, W. Li, J. Shi, M. Wang, F. Xie, X. Zhang, P. Wang, Y. Zhao, N. Li, N. Yang, W. Dong, S. Hu, C. Zeng, W. Zheng, B. Hao, L. W. Hillier, S. P. Yang, W. C. Warren, R. K. Wilson, M. Brandstrom, H. Ellegren, R. P. Crooijmans, J. J. van der Poel, H. Bovenhuis, M. A. Groenen, I. Ovcharenko, L. Gordon, L. Stubbs, S. Lucas, T. Glavina, A. Aerts, P. Kaiser, L. Rothwell, J. R. Young, S. Rogers, B. A. Walker, A. van Hateren, J. Kaufman, N. Bumstead, S. J. Lamont, H. Zhou, P. M. Hocking, D. Morrice, D. J. de Koning, A. Law, N. Bartley, D. W. Burt, H. Hunt, H.H.Cheng, U. Gunnarsson, P. Wahlberg, L. Andersson, K. Institutet, E. Kindlund, M. T. Tammi, B. Andersson, C. Webber, C. P. Ponting, et al. 2004. A genetic variation map for chicken with 2.8 million single-nucleotide polymorphisms. Nature 432:717-722.

[46] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classiers. Proceedings of the Fifth Annual Workshop on Computational Learning, 1992.

[47] Vladimir N. Vapnik, The Nature of Statistical Learning Theory. Springer, 1995

[48] Henderson, C. R. 1984. Application of linear models in animal breeding. Univ. of Guelph, Guelph, ON, Canada.

[49] Hayes, B., J. Laerdahl, D. Lien, A. Adzhubei and B. Hoyheim, 2004 Large scale discovery of single nucleotide polymorphism (SNP) markers in Atlantic Salmon (Salmo salar). AKVAFORSK, Institute of Aquaculture Research

[50] Hastie, T., Tibshirani, R., Friedman, J. Elements of Statistical Learning: Data Mining, Inference and Prediction, (2001) , NY Springer-Verlag.

[51] Haley, C. S., and S. A. Knott, 1992 A simple regression method for mapping quantitative trait loci in line crosses using flanking markers. Heredity 69: 315324.

[52] Knapp, S. J., 1991: Using molecular markers to map multiple quantitative trait loci: models for back-cross, recombinant inbred, and doubled haploid progeny. Theor. Appl. Genet. 81, 333338.

[53] Broman, K. W., and T. P. Speed, 2002 A model selection approach for identification of quantitative trait loci in experimental crosses. J. R. Stat. Soc. B 64: 641656.

[54] Kao, C. H., Z-B. Zeng and R. D. Teasdale, 1999 Multiple interval mapping for quantitative trait loci. Genetics 152: 12031216.

[55] Carlborg, O., L. Andersson and B. Kinghorn, 2000 The use of a genetic algorithm for simultaneous mapping of multiple interacting quantitative trait loci. Genetics 155: 20032010.

[56] Rafal Kustra. Statistics of Data Mining course. Spring 2008.

[57] Ball, R. D., 2001 Bayesian methods for quantitative trait loci mapping based on model selection: approximate analysis using the Bayesian information criterion. Genetics 159: 13511364.

[58] Piepho, H. P., and H. G. Gauch, JR., 2001 Marker pair selection for mapping quantitative trait loci. Genetics 157: 433444.

[59] Bogdan, M., J. K. Ghosh and R. W. Doerge, 2004 Modifying the Schwarz Bayesian information criterion to locate multiple interacting quantitative trait loci. Genetics 167: 989999.

[60] Kadane, J. B., and N. A. Lazar, 2004 Methods and criteria for model selection. J. Am. Stat. Assoc. 99: 279290.

[61] Chipman, H., E. I. Edwards and R. E. Mcculloch, 2001 The practical implementation of Bayesian model selection, pp. 65116 in Model Selection, edited by P. LAHIRI. Institute of Mathematical Statistics, Beachwood, OH.

[62] Geyer, C. J., 1992 Practical Markov chain Monte Carlo. Stat. Sci. 7:473-511.

[63] Green, P. J., 1995 Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. Biometrika 82:711-732.

[64] Satagopan, J. M., B. S. Yandell, M. A. Newton and T. C. Osborn, 1996 Markov chain Monte Carlo approach to detect polygene loci for complex traits. Genetics 144: 805816.

[65] Heath, S. C., 1997 Markov chain Monte Carlo segregation and linkage analysis for oligogenic models. Am. J. Hum. Genet. 61: 748760.

[66] Gaffney, P. J., 2001 An efficient reversible jump Markov chain Monte Carlo approach to detect multiple loci and their effects in inbred crosses. Ph.D. Dissertation, University of Wisconsin, Madison, WI.

[67] Xu, S., 2003 Estimating polygenic effects using markers of the entire genome. Genetics 163: 789801.

[68] Zhang, M., K. L. Montooth, M. T. Wells, A. G. Clark and D. Zhang, 2005 Mapping multiple quantitative trait loci by Bayesian classification. Genetics 169: 23052318.

[69] Breiman, L., 1995 Better subset selection using the nonnegative garotte. Technometrics 37: 373384.

[70] Tibshirani, R., 1996 Regression shrinkage and selection via the lasso. J. R. Stat. Soc. 58: 267288.

[71] Whittaker, J. C., R. Thompson and M. C. Denham, 2000 Marker-assisted selection using ridge regression. Genet. Res. 75: 249252.

[72] Gianola, D., M. Perez-enciso and M. A. Toro, 2003 On marker-assisted prediction of genetic value: beyond the ridge. Genetics 163: 347365.

[73] Gianola D., Fernando R. L., Stella A. (2006) Genomic-assisted prediction of genetic value with semiparametric procedures. Genetics, 173, 17611776.

[74] J.C.M. Dekkers, and F. Hospital. 2002. Utilization of molecular genetics in genetic improvement of plants and animals. Nature Reviews: Genetics 3: 22-32.

[75] Gonzlez-Recio, O., D. Gianola, N. Long, K. A. Weigel, G. J. M. Rosa et al., 2008 Non-parametric methods for incorporating genomic information into genetic evaluations: an application to mortality in broilers. Genetics 178: 23052313.

[76] Ruczinski, I., Kooperberg, C. and LeBlanc, M. (2003). Logic Regression. Journal of Computational and Graphical Statistics, 12, 475-511.

[77] Frank, I. E. and Friedman, J. H. (1993) A statistical view of some chemometric regression tools. Technometrics, 35, 109-148.

[78] Fan, J. and Li, R. (2001) Variable selection via nonconcave penalized likelihood and its oracle properties. J. Am.Statist. Ass., 96, 13481360.

[79] Zou H and Hastie T. Regularization and variable selection via the elastic net. J R Stat Soc Ser B Stat Methodol 67: 301-320, 2005.

[80] Massey, W.F., 1965. Principal components regression in exploratory statistical research. J. Amer. Statist. Assoc. 60, 234246.

[81] Rasmussen, C.-E. and Williams, C.K.I. 2006. Gaussian Processes for Machine Learning, MIT Press. IEEE Trans. on Inform. Theory, 21:438440.

[82] A. J. Smola and B. Schlkopf, A Tutorial on Support Vector Regression, London, U.K.: Royal Holloway College, NeuroCOLT Tech. Rep. TR 1998-030, 1998.

[83] Ein-Dor L, Kela I, Getz G, Givol D, Domany E. Outcome signature genes in breast cancer: is there a unique set? Bioinformatics 2004;21:1718.

[84] Ruczinski, I., Kooperberg, C. and LeBlanc, M. (2004). Exploring Interactions in High-Dimensional Genomic Data: An Overview of Logic Regression, with Applications. Journal of Multivariate Analysis, 90, 178-195.

[85] Zee RYL, Hoh J, Cheng S, Reynolds R, Grow MA, Silbergleit A, Walker K, Steiner L, Zangenberg G, Fernandez-Ortiz A, Macaya C, Pintor E, Fernandez-Cruz A, Ott J, Lindpaintner K. 2002. Multi-locus interactions predict risk for post-PTCA restenosis: an approach to the genetic analysis of common complex disease. Pharmacogenomics J2:197201.

[86] Kooperberg, C., and I. Ruczinski, 2005 Identifying interacting SNPs using Monte Carlo logic regression. Genet. Epidemiol. 28: 157170.

[87] Moore, J. H. and Williams, S. M. (2002). New strategies for identifying gene-gene interactions in hypertension. Ann Med.

[88] Schwender, H. Minimization of Boolean Expressions Using Matrix Algebra. Technical Report, SFB 475, 2006, Department of Statistics, University of Dortmund, Germany.

[89] Schwender, H., Ickstadt, K., 2007. Identification of SNP interactions using logic regression. Biostatistics doi:10.1093/biostatistics/kxm024.

[90] E. Frank and e. al, "Weka [http://www.cs.waikato.ac.nz/ml/weka/]", The University of Waikato, 2002.

[91] Pirooznia M, Yang JY, Yang MQ, Deng Y: A comparative study of different machine learning methods on microarray gene expression data. BMC Genomics 2008, 9(Suppl 1):S13.

[92] Troyanskaya O, Cantor M, Sherlock G, Brown P, Hastie T, Tibshirani R, Botstein D, Altman RB: Missing value estimation methods for DNA microarrays. Bioinformatics 2001, 17(6):520-525.

[93] Christopher J.C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, 1998.

[94] Oba, S., Sato, M., Takemasa, I., Monden, M., Matsubara, K., Ishii, S. (2003) A Bayesian missing value estimation method for gene expression profile data. Bioinformatics, 19, 20882096.

[95] Nguyen DV, Wang N, Carroll RJ: Evaluation of missing value estimation for microarray data. Journal of Data Science 2004, 2:347-370.

[96] Jornsten R, Wang HY, Welsh WJ, Ouyang M: DNA microarray data imputation and significance analysis of differential expression. Bioinformatics 2005, 21(22):4155-4161.

[97] Feten G, Almy T, Aastveit AH: Prediction of missing values in microarray and use of mixed models to evaluate the predictors. Stat Appl Genet Mol Biol 2005, 4():Article10.

[98] Kononenko, I.: Comparison of Inductive and Naive Bayesian Learning Approaches to Automatic Knowledge Acquisition. Current Trends in Knowledge Acquisition. IOS Press (1990)

[99] Meuwissen, T. H., B. J. Hayes and M. E. Goddard, 2001 Prediction of total genetic value using genome-wide dense marker maps. Genetics 157: 18191829.

[100] A. P. W. de Roos, C. Schrooten, E. Mullaart, M. P. L. Calus, and R. F. Veerkamp; Breeding Value Estimation for Fat Percentage Using Dense Markers on Bos taurus Autosome 14; Journal of Dairy Science, October 1, 2007; 90(10): 4821 - 4829.

[101] Long, N., D. Gianola, G. J. M. Rosa, K. Weigel and S. Avendano, 2007 Machine learning classification procedure for selecting SNPs in genomic selection: application to early mortality in broilers. J. Anim. Breed. Genet. 124(6): 377389

[102] Raychaudhuri S, Stuart JM, Altman RB. Principal components analysis to summarize microarray experiments: application to sporulation time series. Pac.Symp. Biocomput.2000, 455-466 (2000).

[103] Breiman L, Friedman J, Olshen R, Stone C, 1984. Classification and Regression Trees, Wadsworth and Brooks Pacic Grove CA.

[104] Efron B, Hastie T, Johnstone T, Tibshirani R. (2004) Least angle regression. Annals of Statistics, 32(2):407-451.

[105] Kim S, Xing E. (2008) Feature Selection via Block-Regularized Regression. Proceedings of the 24th Annual Conference on Uncertainty in Artificial Intelligence (UAI-08).

[106] Barrett JC, Fry B, Maller J, Daly MJ. (2005) Haploview: analysis and visualization of LD and haplotype maps. Bioinformatics.

[107] Gabriel, S.B., Schaffner, S.F., Nguyen, H., Moore, J.M., Roy, J., Blumenstiel, B., Higgins, J., Defelice, M., Lochner, A., Faggart, M., et al. (2002) The structure of haplotype blocks in the human genome. Science, 296, 22252229.

[108] Wang, N., Akey, J.M., Zhang, K., Chakraborty, R., Jin, L. (2002) Distribution of recombination crossovers and the origin of haplotype blocks: the interplay

of population history, recombination, and mutation. Am. J. Hum. Genet., 71, 12271234.