Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

ABSTRACT

TYPE-1 DIABETES RISK PREDICTION USING MULTIPLE KERNEL LEARNING

by
Paras Garg

This thesis presents an analysis of multiple kernel learning (MKL) for type-1 diabetes risk prediction. MKL combines different models and representation of data to find a linear combination of these representations of the data. MKL has been successfully been implemented in image detection, splice site detection, ribosomal and membrane protein prediction, etc. In this thesis, this method was applied for Genome-wide association study (GWAS) for classifying cases and controls.

This thesis has shown that combined kernel does not perform better than the individual kernels and that MKL does not select the best model for this problem. Also, the effect of normalization on MKL as well as risk prediction has also been analyzed.

TYPE-1 DIABETES RISK PREDICTION USING MULTIPLE KERNEL LEARNING

by
Paras Garg

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Bioinformatics

Department of Computer Science

May 2010

APPROVAL PAGE

TYPE-1 DIABETES RISK PREDICTION USING MULTIPLE KERNEL LEARNING

Paras Garg

Dr. Usman Roshan, Thesis Advisor — Date
Assistant Professor of Computer Science, NJIT

Dr. Jason Wang, Committee Member — Date
Professor of Computer Science, NJIT

Dr. Zhi Wei, Committee Member — Date
Assistant Professor of Computer Science, NJIT

BIOGRAPHICAL SKETCH

Author: Paras Garg

Degree: Master of Science

Date: May 2010

Undergraduate and Graduate Education:

- Master of Science in Bioinformatics,
  New Jersey Institute of Technology, Newark, USA, 2010

- Bachelor of Science in Biotechnology,
  Amity University, Noida, India, 2008

Major: Bioinformatics

This thesis is dedicated to my beloved family

To my father and mother who hold the family together

To my sister, who always cared about me and give me strength

To my grandfather and grandmother, who always loved me
I can always feel your blessing

To my best friend Mahesh Chemudupati
Your friendship is the most precious thing I ever had

I'd like to share every shred of my joy and happiness with you

ACKNOWLEDGMENT

I gratefully thank my advisor, Dr. Usman Roshan, for his guidance and encouragement throughout my one-year master's study at the New Jersey Institute of Technology. It has been an invaluable opportunity for me to work at the Bioinformatics Lab under his direction. I believe that what I have learned from him will significantly benefit my future career. Special thanks to my dissertation committee members, Dr Jason Wang and Dr Zhi Wei for their guidance and encouragement. I also thank the department chair, Dr. Narain Gehani, and all other faculty members for their support.

This project would not have completed without the help of System administrators. I would like to thank Kevin Walsh, Douglas Eadline, Gedaliah Wolosh and all other people at system administrator for providing service for Kong and AFS systems. I would like to single out David Perel for his help and support for even small problem with these systems. He has always been eager to listen to the problem patiently, giving priority to it.

I would like to thank my good colleagues and friends, Suresh Solaimuthu and Satish Chikkagoudar for their support of my studies. They have helped time to time with my thesis and encouraged me to achieve my goal. I am grateful to Jay Patel, Meeral Prasad, Maria Somoza, Wei Wang, Raghav Sharma, Ozgur Akcan, David Panek and Young Jae for their constant support and boosting my moral time to time.

Most of all, I am grateful for the constant support, understanding, patience, and trust of my parents, my sister Nidhi Agarwal, without whom none of this work would have been possible.

Thanks to all for helping me complete this thesis.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

GWA	Genome Wide Association
SNP	Single Nucleotide Polymorphism
WTCCC	Wellcome Trust Case Control Consortium
χ^2	Chi Square
SVM	Support Vector Machines
W	Discriminant value
C	Co-efficient for error
MKL	Multiple Kernel Learning
RBF	Radial Basis Function (Gaussian)
Γ	Gamma (for RBF kernel)
D	Degree for (Polynomial kernel)

CHAPTER 1

INTRODUCTION

1.1 Objective

The objective of this thesis is to analyze the performance of multiple kernel learning for Genome wide association study (GWAS) to predict the risk of Type 1 diabetes.

For evaluating the performance, training/testing study was conducted with 10 random splits and Linear, Gaussian and Polynomial kernels were used as base kernels. The classification accuracy and ability of MKL for feature and model selection have been reviewed in this thesis.

1.2 Background Information

Genetic variation—differences in the coding and non-coding portions of DNA causes unique phenotypes in the population. It can also contribute to a personalized susceptibility to disease (Roukos, 2008). The current strategy for revealing the genetic basis of disease susceptibility is to carry out a genome-wide association study (GWAS) with a million or more single nucleotide polymorphisms (SNPs) that capture most of the common variation in the human genome (Moore, 2010). Exhaustive analysis of human SNPs has led to the identification of interesting SNP markers for certain disorders.

1.2.1 GWAS and its Applications

GWA studies were made possible by the sequencing of the human genome using High through put analysis and next generation sequencing (Wellcome Trust Case Control Consortium, 2007) that discovered millions of common SNPs and documented the

correlation structure or linkage disequilibrium of the alleles at those loci. The GWA Study by WTCCC is a case-control design in which allele frequency in patients is compared to the disease free group. WTCCC provided the associated SNPs for common diseases such as diabetes type 1, type II, bipolar disorder, etc.

Besides identifying genes influencing disease susceptibility or phenotypic variation, another often suggested utility of GWAS is that these discoveries will facilitate implementation of personalized medicine, in which preventive and therapeutic interventions for complex diseases are tailored to individuals based on their genetic profiles. Personalized medicine already exists for monogenetic disorders such as Huntington disease, phenylketonuria (PKU) and hereditary forms of cancer, in which genetic testing is the basis for informing individuals about their future health status and for deciding upon specific, often radical interventions such as lifetime dietary restrictions and preventive surgery. (Janssens & Duijn, 2008).

1.2.2 Challenges with GWAS for Common Complex Diseases

The genetic origin of common complex diseases differs essentially from that of monogenic disorders. Unlike monogenic disorders, such as Huntington disease, PKU and hereditary cancers, complex diseases result from the joint effects of multiple genetic and environmental causes, with each factor having only a minor contribution to the occurrence of disease (Janssens & Duijn, 2008).

Recent studies have reasoned the low predictive value of a larger number of multiple weak susceptibility variants. First, when multiple genes are considered simultaneously, one typically finds that all individuals in a population carry at least one or more risk genotypes, even those persons with a lower than average risk of disease.

Second, the more the risk genotypes, the higher the risk of disease, but substantial variation in disease risk may be seen between individuals with the same number of risk genotypes resulting from differences in effect sizes between risk genotypes.

One of the paradigms in complex genetics is that the genetic prediction of common diseases can be substantially improved if genetic variants with strong effects are identified, either on their own or in interaction with other variants or with environmental factors, i.e. gene–gene or gene–environment interaction. Discovering complete causal mechanisms of common diseases implies the identification of specific combinations of causal factors among all possible combinations, namely identifying those combinations that inevitably lead to disease.

1.3 SNP Analysis

One of the ways of identifying the combination of factors (SNPs) associated with the common disease is to rank them according to their prediction risk score under some assumption or model. The most commonly used model is Chi square which is basically a probability based method under some threshold.

The chi-square statistic has also been referred to as genotypic 2 degree-of-freedom test. Define six random variables each of which is binomially distributed Xi ~ B(n; pi) where n is the total number of subjects and pi is the probability of success for Xi. Each of these corresponds to the number of case or control subjects with 0, 1, or 2 copies of the allele of interest. The expected value of each Xi is given by E(Xi) = npi. It can then be shown that the statistic below follows the chi-square distribution with 2 degrees of freedom. This is called the chi-square statistic.

$$\chi^2 = \sum_{i=1}^{6} \frac{(c_i - e_i)^2}{e_i} \tag{1.1}$$

Table 1.1 Contingency Table for SNP Data for Case and Controls

	0	1	2
Case	c_1 (X_1)	c_2 (X_2)	c_3 (X_3)
Control	c_4 (X_4)	c_5 (X_5)	c_6 (X_6)

To apply this statistic for detecting SNPs from associated regions, let the disease type be denoted by the random variable D and genotype by G. If it is assumed that these are independent then P(D;G) = P(D)P(G). These are easy to calculate from counts in the contingency table. For example,

$$P(G = 0) = \frac{(c1 + c4)}{n} \text{ and, } P(D = \text{case}) = \frac{(c1 + c2 + c3)}{n}$$

Similarly, the expected values of each Xi can be calculated under the null hypothesis and consequently the chi-square statistic. For example, the expected value of X1 is given by,

$$E(X_1) = P(X = \text{case})P(Y = 0)n = \frac{(c_1 + c_2 + c_3)(c_1 + c_4)}{n}$$

under the null hypothesis. The corresponding p-values can be obtained by referring to the chi-square distribution with 2 degrees of freedom. SNPs with the least p-values deviate from the independence assumption and therefore are of interest.

CHAPTER 2

MACHINE LEARNING

2.1 Support Vector Machines

Support vector machines (SVMs) have exhibited superb performance in binary classification tasks. Intuitively, SVM aims at searching for a hyperplane that separates the two classes of data with largest margin (the margin is the distance between the hyperplane and the point closest to it). (Vapnik, 1998; Li, Zhang, & Ogihara, 2004). Let the sample be $X = \{ x_i, \square(x_i)\}$ where $y_i = +1$ if $x_i \in C1$ and $y_i = -1$ if $x_i \in C2$. (Alpaydin, 2004).

The equation of hyperplane is:

$$w^T x^t + b = 0 \tag{2.1}$$

Hence, for optimal hyperplane, the hyperplane must best separate the instance with some distance away (margin).

$$w^T x_i + b \geq +1 \text{ for } y_i = +1 \tag{2.2}$$

It can be rewritten as

$$y_i (w^T x_i + b) \geq +1 \tag{2.3}$$

The two variables, w^T and w0, can be calculated by using

$$\frac{y_i (w^T x_i + b)}{\|w\|} \geq \rho \tag{2.4}$$

ρ is the margin which is to be maximized for optimal hyperplane. In other words, $\|w\|$ has to be minimized. This is defined in the paper by Vapnik 1995; Cotes and Vapnik 1995).

$$\text{Min } \tfrac{1}{2} \|w\|^2 \text{ subject to } y_i (w^T x_i + b) \geq +1 \tag{2.5}$$

In finding the optimal hyperplane, this optimization problem is converted to a form whose complexity depends upon the number of training instances and not on the dimension d. The primal is converted to a new formulation using the Langrange multipliers.

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, i = 1, \ldots, l. \end{aligned} \tag{2.6}$$

where, ξ is error for soft margin of the hyperplane.

Its dual is:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} - \mathbf{e}^T\boldsymbol{\alpha} \\ \text{subject to} \quad & \mathbf{y}^T\boldsymbol{\alpha} = 0, \\ & 0 \leq \alpha_i \leq C, \qquad i = 1, \ldots, l, \end{aligned} \tag{2.7}$$

Kernel Trick

If a problem is non linear, instead of fitting a non linear model, the problem can be mapped to a new space by doing a non linear transformation using suitably chosen basis functions and then use a linear model in this new space. This linear model in new space represents the non linear model in the original space (Taylor & Cristianini, 2004).

The decision function is

$$\operatorname{sgn}\left(\sum_{i=1}^{l} y_i\alpha_i K(\mathbf{x}_i, \mathbf{x}) + b\right). \tag{2.7}$$

2.2 Multiple Kernel Learning (MKL)

Basic algebraic operations such as addition, multiplication and exponentiation preserve the key property of positive semi-definiteness, and thus, allow a simple but powerful algebra of kernels (Lanckriet et al. 2004). For example, given two kernel functions K_1 and K_2, inducing the embeddings $\varphi_1(x)$ and $\varphi_2(x)$, respectively, it is possible to define the kernel $K = K_1 + K_2$, inducing the embedding $\varphi(x) = [\varphi_1(x), \varphi_2(x)]$. Of even greater interest, parameterized combinations of kernels can be considered. In particular, given a set of kernels $K = \{K_1, K_2, \ldots, K_m\}$, the linear combination can be formed.

$$\mathbf{k}(\mathbf{x}i,\mathbf{x}j) = \sum_{k=1}^{K} \beta_k \mathbf{k}_k(\mathbf{x}i,\mathbf{x}j) \tag{2.8}$$

with $\beta_k \geq 0$ and $\sum_{Kk=1} \beta_k = 1$, where each kernel $\mathbf{k}_k$ uses only a distinct set of features.

In 2004, Lanckriet et al. have shown that multiple kernel learning can improve the performance of the classifier over single kernel. All the kernels can also involve different kernel functions such as Gaussian or polynomial kernel using different parameters. Within this framework, the problem, the problem of data representation is transferred to the choice of βk.

The values of the coefficients α, β and b can be obtained by solving the dual of the following optimization problem (Schölkopf & J. Smola, 2002):

$$\begin{aligned}
\min \quad & \frac{1}{2}\left(\sum_{k=1}^{K} \|\mathbf{w}_k\|_2\right)^2 + C\sum_{i=1}^{N}\xi_i \\
\text{w.r.t.} \quad & \mathbf{w}_k \in \mathbb{R}^{D_k}, \xi \in \mathbb{R}^N, b \in \mathbb{R}, \\
\text{s.t.} \quad & \xi_i \geq 0 \text{ and } y_i\left(\sum_{k=1}^{K}\langle \mathbf{w}_k, \Phi_k(\mathbf{x}_i)\rangle + b\right) \geq 1 - \xi_i, \ \forall i = 1,\ldots,N
\end{aligned} \tag{2.9}$$

Bach et al (2004) derived this dual formulation and wrote equivalently:

$$\begin{aligned} \min \quad & \gamma \\ \text{w.r.t.} \quad & \gamma \in \mathbb{R}, \alpha \in \mathbb{R}^N \\ \text{s.t.} \quad & \mathbf{0} \le \alpha \le \mathbf{1}C, \ \sum_{i=1}^{N} \alpha_i y_i = 0 \\ & \underbrace{\frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{k}_k(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{N} \alpha_i}_{=:S_k(\alpha)} \le \gamma, \ \forall k = 1, \ldots, K \end{aligned} \tag{2.10}$$

It the above optimization problem, the $\ell 1$-norm of β is constrained to one, while one is penalizing the $\ell 2$-norm of wk in each block k separately. The idea is that $\ell 1$-norm constrained or penalized variables tend to have sparse optimal solutions, while $\ell 2$-norm penalized variables do not (e.g., Rätsch, 2001, Chapter 5.2). Thus the above optimization problem offers the possibility to find sparse solutions on the block level with non-sparse solutions within the blocks.

Sonnenburg et al. (2006) used semi definite programming (SLIP) to find the optimal solution to the problem:

$$\begin{cases} \max\limits_{\theta,\beta} & \theta \\ \text{s.t.} & \sum_{k=1}^{K} \beta_k = 1 \\ \text{and} & \sum_{k=1}^{K} \beta_k S_k(\alpha) \ge \theta \end{cases}$$

$$\text{with } S_k(\alpha) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j \mathbf{k}_k(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{N} \alpha_i \tag{2.11}$$

CHAPTER 3

METHODS

3.1 Problem Statement

MKL has been successfully implemented for classification problems such as image detection, splice site detection, prediction of structure and function of proteins etc. The objective of this thesis was to analyze the strength of MKL in genomics for risk prediction for type-1 diabetes. This paper focuses on answering the following questions:

1. Can MKL produce a kernel that has significantly higher prediction accuracy than the base kernels?
2. Can MKL be used to identify the most significant features/models based on the weights β?

3.2 Dataset

The Wellcome Trust Case Control Consortium (WTCCC, 2007) provides two set of controls and on set of cases for type 1 diabetes. Individuals, whose genotypes were included in the study were living within England, Scotland and Wales ('Great Britain') and the vast majority had self-identified themselves as white Europeans.

Same method was used, for filtering the SNPs that where regarded problematic by the WTCCC. This left with 1480 individual from British Birth Cohort, 1458 from UK Blood Service Control Group and 1963 cases for type 1 diabetes with 422,006 SNPs. This dataset was converted to encoded matrix of 0, 1 and 2's by standard encoding (Price et. al). In this thesis, 0, 1 and 2 represents two, one and zero copies of risk alleles.

Raw data

	A/T	C/T	C/G	A/C	C/G
Cases	AA	TT	CG	CC	GG
	AT	TT	CG	CC	GG
Controls	TT	CT	GG	AC	CC
	TT	TT	GG	AA	CC

↓

Numerical conversion according to the copies of risk alleles

A/T	C/T	C/G	A/C	C/G
0	2	1	2	2
1	2	1	2	2
2	1	2	1	0
2	2	2	0	0

↓

Chi Square ranking

	C/G	A/T	A/C	C/G	C/T
Cases	2	0	2	1	2
	2	1	2	1	2
Controls	0	2	1	2	1
	0	2	0	2	2

↓

Base Kernel (Linear here)

Each cell represents the dot product of two vectors

	Case 1	Case 2	Control 1	Control 2
Case 1	13	13	6	6
Case 2	13	14	8	8
Control 1	6	8	10	10
Control 2	6	8	10	12

↓

Multiple Kernel Learning

Figure 3.1 Toy example to illustrate the process of the experiment. The raw SNP data is converted to 0, 1 and 2 encoding, which is then ranked by Chi square. Various kernels and nuber of features are used in the experiment

3.3 Base Kernels

For MKL, three standard kernel functions: Linear, Polynomial (Degree=1, 2) and Gaussian (γ =1.2, 2, 5) were used.

Linear Kernel: $k(x, x') = \langle x, x' \rangle$

Polynomial Kernel: $k(x, x') = \langle x, x' \rangle^d$

(d = 1, 2)

RBF Kernel: $k(x, x') = \exp\left(\frac{\|x \quad x'\|^2}{2\sigma^2} \right)$

(γ =1.2, 2, 5)

3.4 Implementation

Due to large amount of data, the feature selection method was implemented in C program. After ranking according to their p value, top $\chi 2$ ranked 1000 SNPs were selected for this analysis. The command line implementation of MKL was used that is available at http://www.shogun-toolbox.org/. The base kernels were generated using 20, 40, 60, 80, 100, 200, 400, 600, 800 and 1000 top χ^2 ranked SNPs. Each of the kernels was normalized. For normalization, general formatting and data selection, Perl scripts were used.

CHAPTER 4

RESULTS

4.1 MKL and SVMlight

In order to learn a single kernel with MKL classifier (for consistency), the results of Shogun MKL single kernel with SVMlight results has been compared. In (Sonnenburg, 2006), it has been stated that when K (number of kernels) is equal to 1, the MKL problem reduces to original SVM dual. Therefore, to verify this statement, kernel were learned with SVMlight as well as Shogun MKL at C=1, 0.001. In this case, the un-normalized kernels were used.

Table 4.1 Comparison of SVMlight and Shogun MKL

	C	20	40	60	80	100	200	400	600	800	1000
Svmlight linear	1	78.84	79.00	79.25	79.74	79.96	81.26	81.28	80.00	78.58	76.27
MKL Linear Unnormalized	1	78.84	79.02	79.29	79.80	80.02	81.22	81.32	80.43	78.58	76.27
SVM light linear	0.001	77.15	77.78	77.80	78.27	78.43	79.76	80.92	81.16	80.98	80.47
MKL Linear Unnormalized	0.001	77.15	77.78	77.80	78.27	78.45	79.74	80.92	81.16	80.96	80.47

Table 4.1 shows the prediction accuracy for SVMlight and MKL with various numbers of features. It can be clearly seen that both the methods display similar accuracies. Though, MKL solves the same dual as done by SVMlight, there is small variation ($\sim\pm 0.02$) in their accuracies. This variation can be explained due to the fact that MKL uses SLIP to solve the quadratic dual problem and this makes MKL slower than SVMlight during optimization. Hence it is verified that Shogun MKL works same as SVMlight when the number of kernels is equal to 1.

4.2 Comparison of Standard Kernels

Before running MKL, the prediction accuracies standard kernels: Linear, Polynomial and RBF have been compared with various numbers of features (Table 4.2). It can be observed that the linear kernel performs the best among other kernels. The polynomial classifier (degree 2 and 3) shows moderate performance with high number of feature, however, it suffers with less number of features. On the other hand, RBF kernel with gamma=1.2, 2 and 5 have shown poor performance. Though the results for RBF with gamma=1.2 does produce predictive accuracy >70% , it keeps on decreasing with increasing number of SNPs.

Table 4.2 Prediction Accuracies for Various Models and Features (SNPs)

Model /SNPs	20	40	60	80	100	200	400	600	800	1000
Linear	78.84%	79.00%	79.25%	79.74%	79.96%	81.26%	81.28%	80.00%	78.58%	76.27%
Poly (d=2)	71.65%	48.76%	55.85%	56.50%	56.13%	71.69%	75.97%	76.48%	77.05%	77.19%
Poly (d=3)	42.28%	53.22%	52.99%	55.95%	55.62%	72.40%	76.62%	77.33%	77.15%	77.62%
RBF (γ=1.2)	77.68%	71.94%	70.12%	67.46%	63.12%	60.06%	59.88%	59.88%	59.88%	59.88%
RBF (γ=2)	75.56%	69.69%	67.68%	65.30%	61.57%	59.92%	59.88%	59.88%	59.88%	59.88%
RBF (γ=5)	75.23%	69.37%	67.33%	64.91%	61.42%	59.92%	59.88%	59.88%	59.88%	59.88%

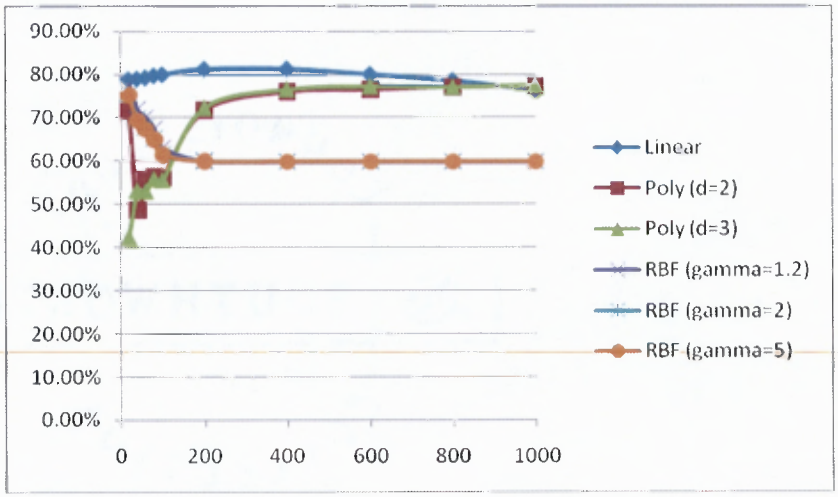


Figure 4.1 Comparison of various models with increasing number of features (SNPs).

As seen in Figure 4.1, the linear kernel displays slight increase in accuracy initially, which marginally drops with low number of SNPs. This is because; with higher number of SNPs there are high chances of adding noise to the data. The polynomial kernel (degree 2 and 3) follow similar line with initial major drop in performance and then it increases to 75%. With RBF kernels, there is gradual decrease in prediction accuracy till features <100 and then it reaches the minimum (59.88%). The discriminant values for RBF kernel suggest that with higher number of features, the discriminant values are all negative for all the subjects; in other words, the data points are on one side of the hyperplane. This means that RBF kernels (gamma=1.2, 2, 5) is unable to classify even a single feature correctly.

As the conclusion of above results, linear kernel for most of part of the experiment as it indicate better performance than polynomial and RBF kernels.

4.3 MKL Performance

In this section, the first part of the objective has been examined; whether MKL performs better than the individual kernels or not. This experiment was divided in two parts:

1. MKL with various standard kernels as base kernels
2. MKL with various numbers of features as base kernels

4.3.1 MKL with Various Standard Kernels as Base Kernels

In this case, linear, polynomial (degree 2 and 3) and RBF (gamma = 1.2, 2 and 5) were uesd as the base kernels for Shogun MKL. As suggested by Sonnenburg, all the kernels were normalized for consistency. The Table 4.3 shows the results of these base kernels compared with linear normalized kernel. In both the cases, the value of c was equal to 1.

Table 4.3 Comparison MKL with Linear, Polynomial and Gaussian (RBF) as Base Kernel to Linear Normalized Kernel

	20	40	60	80	100	200	400	600	800	1000
MKL L+P2+P3+G(1.2) +G(2)+G(5)	79.47	78.49	78.53	78.68	78.72	79.63	80.14	80.53	80.53	79.84
Linear Normalized	78.68	79.33	79.43	79.47	79.71	80.43	81.32	81.12	80.57	80.37

MKL does not provide better prediction accuracy over individual kernel. The only improvement was observed with 20 features. In all other cases, MKL predictions were low by 0.5% to 1% in comparison with linear kernel.

4.3.2 MKL with Various Numbers of Features as Base Kernels

Since MKL with various standard kernels failed to provide any improvement, various numbers of features were used as base kernels for MKL. In this case, the effect of normalization on prediction accuracies has also been analyzed. The value of c was equal to 1 and 0.001.

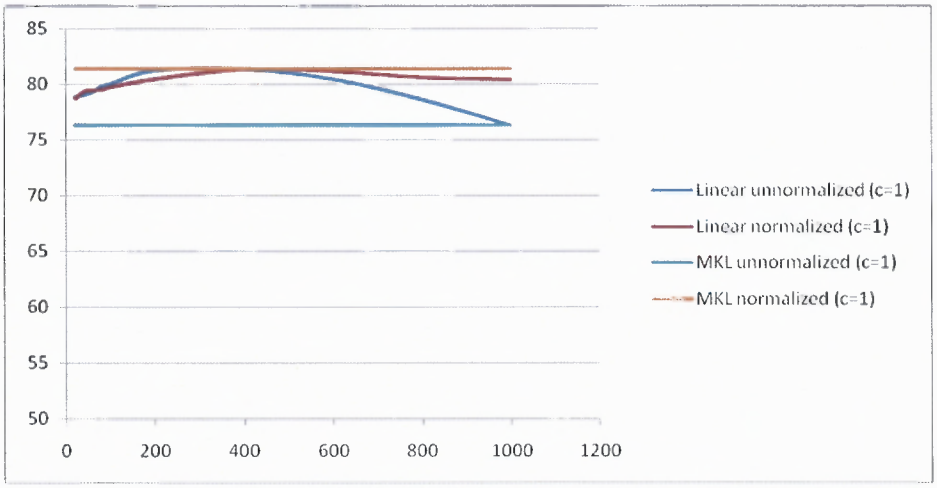
Table 4.4 Comparison of MKL with Various Number Features to Individual Linear Kernels (Normalized and un-normalized).

	C	20	40	60	80	100	200	400	600	800	1000	MKL
Linear Unnormalized	1	78.84	79.02	79.29	79.80	80.02	81.22	81.32	80.43	78.58	76.27	76.27
Linear Normalized	1	78.68	79.33	79.43	79.47	79.71	80.43	81.32	81.12	80.57	80.37	81.36
Linear Unnormalized	0.001	77.15	77.78	77.80	78.27	78.45	79.74	80.92	81.16	80.96	80.47	80.47
linear normalized	0.001	59.87	59.87	59.87	59.87	59.87	59.87	59.87	59.87	59.87	59.87	59.87

For un-normalized kernel, MKL was unable to predict better than the individual kernel. For C=1, the MKL produces the worst kernel among all the kernels. For c=0.001, MKL performs lower than the best performing individual kernel. Normalization has major impact on the prediction accuracy. With C=1, MKL performs as well as the best

performing kernel (Figure 4.2). However, with c=0.001, the accuracy for various number of features is ~60%. This is probably because, with normalization, all the data points reduce to a unit sphere which causes loss of the information.

a)



b)

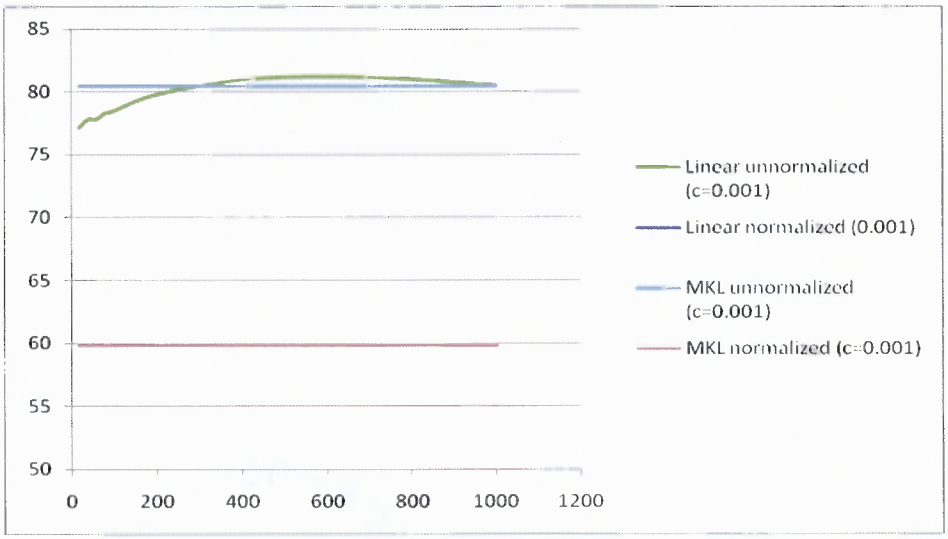


Figure 4.2 Comparision of MKL and linear kernel with (c=1 and 0.001). a) Nomalized b) Unnormalized.

4.4 Feature and Model Selection

In this section, second objective has been examined; whether weights β determined my MKL can be used to select most significant features. According to the hypothesis, MKL must provide higher weight to the most significant feature. A similar study was conducted by Suard et.al. (2007), where they used MKL for pattern recognition using various representations of image such as pixel value, gradient norm, wavelet and histograms of gradients. They concluded that MKL provide higher weight to the most important representation. Similar study has been conducted in this thesis for GWA study.

4.4.1 Weights β and Model selection

For model selection, linear, polynomial (degree = 2,3) and RBF (gamma = 1.2, 2, 5) were used. According to the null hypothesis, linear kernel must be weighted higher by MKL over other kernels as it performs better than other kernels, individually. The value of C was equal to 1.

Table 4.5 Weights β from MKL with Various Models as Base Kernel

SNPs/Models	Linear	Poly (d=2)	Poly (d=3)	RBF (1.2)	RBF (2)	RBF (5)
20	0	0	0.044768	0.921149	0	0.034082
40	0	0	0.066979	0.847248	3.2E-06	0.085769
60	0	0	0.078481	0.791593	7.72E-05	0.129849
80	0	0	0.083845	0.696262	0.000198	0.219695
100	0	0	0.100046	0.613535	0.000432	0.285987
200	0	8.89E-07	0.134695	0.388294	0.004594	0.472415
400	6.67E-07	1.78E-06	0.167613	0.271271	0.275391	0.285722
600	3.33E-07	1.11E-06	0.18945	0.266188	0.269227	0.275135
800	2.22E-07	1.11E-06	0.221311	0.256693	0.257779	0.264216
1000	1.11E-07	8.89E-07	0.255235	0.246111	0.246715	0.251938

Table 4.5 shows the weights β assigned by MKL. For lower number of features, it can be observed that MKL provides 0 weight to linear kernel, which has the best performance individually. On the other hand, RBF (gamma=1.2) which has the lowest

performance, gets the highest weight. With higher number of features, RBF kernels and polynomial kernel with degree 3 are weighted almost equally while linear kernel still recieves the lowest weight.

4.4.2 Weights β and Feature selection

In this case, linear kernel was used with features 20,40, 60, 80, 100, 200, 400, 600, 800 and 1000. Comparison of normalized as well as un-normalized kernel with c=1 and 0.001 has been reported in this section.

Table 4.6 Weights β from MKL with Various Features (SNP) as Base Kernel

	Linear Unnormalized	Linear Normalized	Linear Unnormalized	linear normalized
C	1	1	0.001	0.001
20	0	0.21076	0	0.566349
40	0	0.021115	0	0.24646
60	0	0	0	0.16797
80	0	1.50E-06	0	0.016291
100	0	0.004116	0	0.002712
200	0	3.70E-06	0	7.12E-05
400	0	0	0	4.31E-05
600	0	0	0	3.73E-05
800	0	0	0	3.39E-05
1000	1	0.764003	1	3.23E-05

For un-normalized kernel, c=1 and 0.001, MKL weights the kernel from 1000 features the highest i.e., 1, which means that this kernel is the only important kernel among all. When compared with individual kernel (Table 4.6), kernel from 1000 feature has the lowest prediction accuracy. This can be explained with the fact that MKL does have some issue with data ambiguity and data stability with un-normalized kernels.

With normalized kernel, for C=1, 1000 feature kernel still obtained the highest weight, however, it assigned some weight to 20 and 40 feature kernel too. According to the hypothesis, 400 and 600 feature kernel must receive the highest weights as they are more significant than the other features (table 4.6). In this case also, these kernel obtained β as 0.

CHAPTER 5

CONCLUSION

In this thesis, the performance of MKL was analyzed with GWA study for type 1 diabetes. The kernel learned using combination of various base kernels, failed to provide better performance than the individual kernel. Also, MKL was unable to select the most important features and models. Unlike the performance in other classification problems (such as, image recognition, splice site detection and protein function prediction), MKL performed lower than the individual base kernels for GWA study.

Comparison of normalized and un-normalized kernels shows that normalization greatly affects the prediction accuracy as well as MKL performance. When un-normalized kernel with various features was used as base kernels, it assigned weight β as 1 to the kernel from highest number of features. This can be either due to data ambiguity or data instability. Bach et al. (2007) also suggested that in practice normalization leads to bad predictive performance.

APPENDIX A

SCRIPTS FOR FORMATTING THE DATA AND RUNNING SHOGUN MKL

This appendix includes the perl scripts used for formatting of data for the input for SVMlight and Shogun MKL software. The script for running Shogun MKL has alson been provided.

A.1 To Format the Original Data File to SVMLight Format

```
##########################################################################
#Purpose: To format the original data file to SVMLight format
#Input: <Kernel_File> <randclass> <trueclass> <ouput_training_file> <output_test_file>
#Output: Formatted training and testing file
##########################################################################

$kernel=shift; $rand=shift; $true_file=shift; $train_file=shift; $test_file=shift;

#Reading the data file
open(IN, $kernel);
while(<IN>){ chomp $_; @s=split(/\s+/,$_);
for(my $i=0;$i<scalar(@s); $i++){ K[$j][$i]=$s[$i]; }  $j++; }
close IN;
$total=$j; $dim=@s;

#Reading randclass file
open(IN, $rand);  $j=0;
while(<IN>){ chomp $_; @s=split(/\s+/,$_);
for(my $i=0;$i<scalar(@s); $i++){ $train[$j][$i]=$s[$i]; }  $j++;  }
close IN;
$total_train = $j;

#Reading true class file
open(IN, $true_file); $j=0;
while(<IN>){ chomp $_; @s=split(/\s+/,$_);
for(my $i=0;$i<scalar(@s); $i++){ $true[$j][$i]=$s[$i]; }  $j++;}
close IN;

$k=0; $flag=0;
#Separating training and testing IDs using randclass file
for(my $i=0; $i<$total; $i++){
  #flag 1 means the entry found in randclass (training set)
  for(my $j=0; $j<$total_train; $j++){if($train[$j][1]==$i){ $flag=1; }}
  if($flag==0) { $test[$k][0]=$true[$i][0]; $test[$k][1]=$true[$i][1]; $k++; }
  $flag=0;
```

```
}
$total_test=$k;

#Retrieving training data using the training IDs
for($i=0; $i<$total_train; $i++){ for($j=0; $j<$dim; $j++){
$train_kernel[$i][$j]=$K[$train[$i][1]][$j]; } }

#Retrieving test data using the test IDs
for($i=0; $i<$total_test; $i++){  for($j=0; $j<$dim; $j++){
$test_kernel[$i][$j]=$K[$test[$i][1]][$j]; }}

$dim=shift;
open(OUT, ">$train_file");
for(my $i=0; $i<$total_train; $i++){
   $z=$i+1;
     if($train[$i][0]==0) { $cl =  "-1 "; }
   else { $cl =  "$train[$i][0] "; }

  for(my $j=0; $j<$dim; $j++){
   if($j==0){ print OUT $cl; }
    print OUT ($j+1);
    print OUT ":";
    printf OUT "%.5f ", $train_kernel[$i][$rank[$j]];;
    print OUT " ";
  }
  print OUT "\n";
}

close OUT;
print "\n";
open(OUT, ">$test_file");
for(my $i=0; $i<$total_test; $i++){
   $z=$i+1;
   if($test[$i][0]==0) { $cl =  "-1 "; }
   else { $cl =  "$test[$i][0] "; }
  for(my $j=0; $j<$dim; $j++){
   if($j==0){ print OUT $cl; }
   print OUT ($j+1);
   print OUT ":";
   printf OUT "%.5f ", $test_kernel[$i][$rank[$j]];
   print OUT " ";
  }
  print OUT "\n";
}

 close OUT
```

A.2 To Convert the Original SNP Dataset to the Input for Shogun MKL

```
##########################################################################
#Purpose: To convert the original SNP dataset to the input for Shogun MKL
#Input: <Original Data> <randclass> <trueclass> <Formatted Training File> <Formatted
Testing File> <Dimension>
#Output: Formatted Training File, Formatted Testing File and true class file with testlabel
file
##########################################################################

$kernel=shift; $rand=shift; $true_file=shift; $train_file=shift; $test_file=shift;

#Reading the data file
open(IN, $kernel); while(<IN>){ chomp $_; @s=split(/\s+/,$_);
for(my $i=0;$i<scalar(@s); $i++){ $K[$j][$i]=$s[$i]; }  $j++; } close IN;
$total=$j; $dim=@s;
$j=0;

#Reading the randclass file
open(IN, $rand); while(<IN>){  chomp $_; @s=split(/\s+/,$_);
for(my $i=0;$i<scalar(@s); $i++){ $train[$j][$i]=$s[$i]; }  $j++; } close IN;
$total_train = $j;
$j=0;

#Reading the trueclass file
open(IN, $true_file); while(<IN>){  chomp $_; @s=split(/\s+/,$_);
for(my $i=0;$i<scalar(@s); $i++){ $true[$j][$i]=$s[$i]; }  $j++; } close IN;

#Separating training from testing
$k=0; $flag=0;
for(my $i=0; $i<$total; $i++){
 for(my $j=0; $j<$total_train; $j++){  if($train[$j][1]==$i){ $flag=1; }}
 if($flag==0) { $test[$k][0]=$true[$i][0];    $test[$k][1]=$true[$i][1];    $k++; }
 $flag=0;
}
$total_test=$k;

#for(my $i=0; $i<$total_train; $i++){ print "$train[$i][0] $train[$i][1] "; }
#for(my $i=0; $i<$total_test; $i++){ print "$test[$i][0] $test[$i][1] "; }

for($i=0; $i<$total_train; $i++){
 for($j=0; $j<$dim; $j++){ $train_kernel[$i][$j]=$K[$train[$i][1]][$j]; }
 if($train[$i][0]==0){ $train_label[$i]=1;} else {$train_label[$i]=-1;}
}
```

```
for($i=0; $i<$total_test; $i++){ for($j=0; $j<$dim; $j++){
$test_kernel[$i][$j]=$K[$test[$i][1]][$j]; }
 if($test[$i][0]==0){ $test_label[$i]=1;} else {$test_label[$i]=-1;}
}
#Printing the training data in the file
while($dim=shift){
print "\n";
open(OUT, ">$train_file$dim");
for(my $j=0; $j<$dim; $j++){
 $str="";
 for(my $i=0; $i<$total_train; $i++){ $str.= $train_kernel[$i][$j]." "; }
 chop $str; print OUT $str; print OUT "\n";
}
close OUT;

#Printing the testing data in the file
open(OUT, ">$test_file$dim");
for(my $j=0; $j<$dim; $j++){
 $str="";
 for(my $i=0; $i<$total_test; $i++){ $str.= $test_kernel[$i][$j]." "; }
 chop $str; print OUT $str; print OUT "\n";
}
close OUT;
}

#Printing the training and testing labels
open(OUT, ">truelabel"); $temp = join(" ", @train_label); print OUT "$temp\n"; close
OUT;
open(OUT, ">testlabel"); $temp = join(" ", @test_label); print OUT "$temp\n"; close
OUT;
```

A.3. Script to Run Shogun MKL

```
set_labels TRAIN truelabel
clean_features TRAIN
clean_features TEST
set_kernel COMBINED 60

add_kernel 1 LINEAR REAL 20
add_features TRAIN train20
add_features TEST test20
set_kernel_normalization SQRTDIAG

add_kernel 1 LINEAR REAL 20
add_features TRAIN train40
add_features TEST test40
set_kernel_normalization SQRTDIAG

add_kernel 1 LINEAR REAL 20
add_features TRAIN train60
add_features TEST test60
set_kernel_normalization SQRTDIAG

add_kernel 1 LINEAR REAL 20
add_features TRAIN train80
add_features TEST test80
set_kernel_normalization SQRTDIAG

add_kernel 1 LINEAR REAL 20
add_features TRAIN train100
add_features TEST test100
set_kernel_normalization SQRTDIAG


add_kernel 1 LINEAR REAL 20
add_features TRAIN train200
add_features TEST test200
set_kernel_normalization SQRTDIAG

add_kernel 1 LINEAR REAL 20
add_features TRAIN train400
add_features TEST test400
set_kernel_normalization SQRTDIAG

add_kernel 1 LINEAR REAL 20
add_features TRAIN train600
add_features TEST test600
```

```
set_kernel_normalization SQRTDIAG

add_kernel 1 LINEAR REAL 20
add_features TRAIN train800
add_features TEST test800
set_kernel_normalization SQRTDIAG

add_kernel 1 LINEAR REAL 20
add_features TRAIN train1000
add_features TEST test1000
set_kernel_normalization SQRTDIAG

new_classifier MKL_CLASSIFICATION
c 1
train_classifier
output = classify
weights_norm = get_subkernel_weights
```

REFERENCES

Alpaydin, E. (2004). *Introduction to Machine Learning.* Cambridge, Massachusetts: The MIT Press.

Bach, F., Lanckriet, G., & Jordan, M. (2004). Multiple kernel learning, conic duality and the SMO algorithm. *Twenty-first international conference on Machine learning* , ACM.

Janssens, A. C., & Duijn, C. M. (2008). Genome-based prediction of common diseases: advances and prospects. *Human Molecular Genetics, 17* , R166-R173.

Lanckriet, G. R., Deng, M., Cristianini, N., Jordan, M. I., & Noble, W. S. (2004). Kernel-Based Data Fusion and Its Application to Protein Function Prediction in Yeast. *Pacific Symposium on Biocomputing, 9* , 300-311.

Li, T., Zhang, C., & Ogihara, M. (2004). A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics, 20(15)* , 2429–2437.

Moore, J. H., Asselbergs, F. W., & Williams, S. M. (2010). Bioinformatics challenges for genome-wide association studies. *Bioinformatics, 26(4)* , 445 - 455.

Roukos, D. H. (2009). Personal Genomics and Genome-Wide Association Studies: Novel Discoveries but Limitations for Practical Personalized Medicine. *Annals of Surgical Oncology, 16(3)* , 772-773.

Schölkopf, B., & J. Smola, A. (2002). *Learning with kernels: support vector machines, regularization, optimization.* Cambridge: MIT Press.

Sonnenburg, S., Ratsch, G., Schafer, C., & Scholkopf, B. (2006). Large scale Multiple kernel learning. *Journal of Machine Learning Research, 7* , 1531-1565.

Taylor, J. S., & Cristianini, N. (2004). *Kernel Methods for pattern analysis.* Cambridge: Cambridge University Press.

Vapnik, V. (1998). Statistical Learning Theory. *Wiley* .

Wellcome Trust Case Control Consortium. (2007). Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature, 447* , 661-678.