A study of multiple kernel learning for predicting type-1 diabetes from WTCCC genome wide association studies

Paras Garg and Usman Roshan

Abstract— Several recent studies ecent studies of type 1 diabetes prediction from genome wide association studies (GWAS) consider only linear relationships between SNPs. With the kernel trick one can examine non-linear relationships using a linear classifier such as the support vector machine. However, it isn't clear in advance which non-linear kernel to employ. Multiple kernel learning (MKL) provides one solution by finding the best linear combination of different base kernels each representing a different non-linear relationship between SNP genotypes. In this study we set out to explore two questions on the WTCCC type 1 diabetes GWAS. First, can we predict type 1 diabetes with an MKL kernel better than the traditional linear kernel? Second, can we determine the best kernel and the best set of SNPs using MKL coefficients? For the first problem we used a combination of linear, polynomial and Gaussian kernels but found no improvement in risk prediction accuracy over the linear one. In the second problem we compute linear kernels with different set of SNPs and expected the set with the best accuracy to yield a high coefficient. However, this was not the case either.

Index Terms—Use about four key words or phrases in alphabetical order, separated by commas.

1 Introduction

Disease risk prediction from genomic data is a cornerstone problem in medical genetics [1]. Several studies have examined disease risk prediction mostly with logistic regression and significant SNPs selected from genome-wide association studies (GWAS) (CITE). The general approach has been to first split the GWAS subjects into two sets: training and validation. The next step is to select SNPs with p-values under some threshold on just the training data, learn a model on the training data such as logistic regression, and predict case and control status on the validation data. Other studies have replaced the second step with other classifiers such as support vector machine (CITE).

The common thread in these studies is that the classifiers are linear and so they assume a linear relationship between the SNPs. We want to know if utilizing non-linear relationships will lead to higher accuracies. Instead of applying non-linear classifiers that are usually computationally expensive we consider higher dimensional feature spaces that measure non-linearity between SNPs. Fortunately we can do this without having to explicitly compute the new feature space. This is called the kernel trick. We rely on the fact that linear classifiers use the dot product for classification. As long as we can compute the dot product in the higher dimensional space we can build a model in that space and classify case and control there. Kernels allow us to compute dot products in such higher spaces (CITE).

However, it's not clear in advance which kernel to employ. Multiple kernel learning (MKL) provides one solution by finding the linear combination of a specified set of kernels that yields that largest support vector machine margin (CITE). To apply MKL for predicting type 1 diabetes we first rank SNPs in the training set according to chi-square (2-df test) p-values. We select the top 1000, compute different kernels with different SNP sets, learn the best linear combination of the kernels along with the SVM margin, and then predict case and control in the validation set with the new kernel.

In this paper we want to answer two particular 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?

We show that the prediction accuracy with the MKL kernel is about the same as the linear. We don't see a correlation between the kernel with the highest accuracy and the MKL one.

2 Methods

Dataset. For this study we consider the Wellcome Trust Case Control Consortium (WTCCC) type 1 diabetes GWAS (CITE). After following standard quality control steps it contains 1924 case subjects, 2938 controls, and 402532 SNPs.

We used the same method for filtering the SNPs that where regarded problematic by the WTCCC. This left us 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 our case, 0, 1 and 2 represents two, one and zero copies of risk alleles.

- *Paras Garg is with Mt. Sinai Hospital, E-mail: paras@mtsinai.edu.*
- *Usman Roshan is with the Department of Computer Science, New Jersey Institute of Technology, GITC 4400, University Heights, NJ 07102. E-mail: usman@cs.njit.edu.*

Raw data

	A/T	C/T	C/G	A/C	C/G
Cas-	AA	TT	CG	CC	GG
	AT	TT	CG	CC	GG
Con-	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
Cas-	2	0	2	1	2
	2	1	2	1	2
Con-	0	2	1	2	1
	0	2	0	2	2

↓

Linear kernel

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

Example to illustrate conversion of genome-wide association study to numerical format and the linear kernel.

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

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

Polynomial Kernel: (d = 1, 2) $k(x, x') = \langle x, x' \rangle^d$

Gaussian Kernel: (=1.2, 2, 5) $k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$

Implementation. We implemented our own C program for ranking SNPs by chi-square. After ranking the according to their p value, we selected top 1000 X^2 ranked SNPs for our analysis. We used the command line implementation of MKL that is available at http://www.shogun-toolbox.org/. We generated several base kernels each with the top 20, 40, 60, 80, 100, 200, 400, 600, 800 and 1000 X^2 ranked SNPs. Each of the kernels was normalized. We used Perl scripts for normalization, general formatting and data selection.

3 Results

We randomly select 90% of the total subjects in the WTCCC type 1 diabetes GWAS for training and leave the remainder for validation. We generate 10 such random splits.

3.1 MKL and SVMlight

When just the linear kernel is used the MKL problem reduces to original SVM dual (Sonnenburg, 2006). We verify this on our data with different SNP sets and use the SVM-light software package for computing the SVM. We tested Shogun MKL with single kernel at c=1, 0.001.

For consistency, we used un-normalized linear kernels for both the methods. It is demonstrated that the results of MKL with K=1 are consistent at different number of features (SNPs). The small variation (~ ±0.02) in the accuracies in two methods can be explained by the fact that Shogun MKL uses SLIP to solve the quadratic dual problem and this also makes MKL slower than SVMlight. The best performance was achieved at C=1 and features=400 by both the methods. (Table)

3.2 Comparison of Standard Kernels

We compared the results of three standard kernels with different parameters: Linear, Polynomial (d=2,3) and Gaussian (γ= 1.2, 2, 5). The regularization parameter C was set 1 for all the cases. These standard kernels were compared with different number of features: 20, 40, 60, 80, 100, 200, 400, 600, 800, 1000 (Table 3.1 and fig 3.1). The comparison of these standard kernels shows that linear kernel performs better than other kernels. Linear kernel achieves classification accuracy of 78.84% with 20 features and 81.28% with 400 features. Polynomial kernel (d=2,3) showed lower performance with lesser number of features (<200), however showed a great improvement with higher number of features. On the other hand, RBF kernel produced classification accuracy ~75% with 20 features, however it decreases with the increase in number of features. It reaches its minimum of 59.88%. It can be explained by that fact that the discriminant value shows that all the data points in new feature space are one side of the hyperplane.

3.3 MKL Performance

In this section, we examine the first objective and expect MKL to learn a classifier better than the individual kernels. We divided the experiment in two parts:

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

MKL with various standard kernels as base kernels. In this case, we used linear, polynomial (d=2,3) and rbf (γ=1.2, 2, 5) as the base kernel for multiple kernel learning. As suggested by Sonnenburg et al, we normalized all the kernels for data stability. Table 3.2 shows the result of MKL compared with the linear kernel for different number of features. On an average, MKL performed lower than the linear kernel with a difference of ~0.59%. Only when # of features = 20, MKL showed improvement over linear kernel. In all other features, linear kernel had higher classification accuracy.

MKL with various number of features as base kernels. We learned linear kernel with various number of features and used them as base kernels for MKL. The results are compared with normalized as well as un-

normalized kernels and at C=1, 0.001 (Table 3.3 and Figure 3.2). When un-normalized kernels were used with MKL, it performed equal to linear kernel with 1000 features at C=1, 0.001. We achieved highest performance with 20 features which was better than MKL. With normalized kernels at C=1, MKL performs slightly better than its base kernels with an improvement of 0.04% over the linear kernel with 400 features. When C=0.001 was used, the accuracies of all the base kernels were 59.87%. This is probably because normalization brings all the data points to a unit sphere, losing much of the information (see Section).

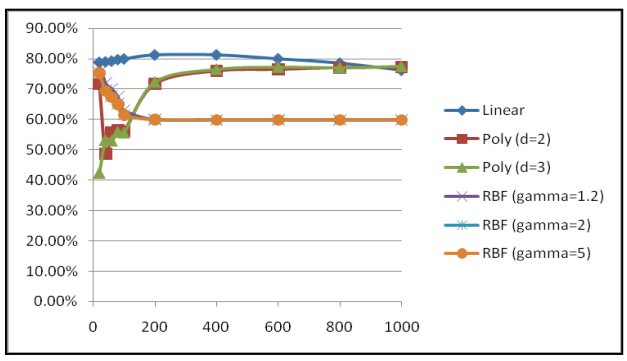
3.4 Feature and Model Selection

In this section, we examine the second objective, to analyze the weights β assigned by MKL to its base kernels and its possible application for feature and model selection. with various numbers of features as base kernels. 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.

Weights β & Model Selection. We obtained the weights assigned by MKL to the standard kernels when we use them as base kernels (Table 3.4). We produced the results for various numbers of features. In most of the cases, MKL assigns higher weight to RBF kernel while lowest weight to linear kernel. Individually, RBF kernel performs the least while linear kernel has best classification accuracy among other models. This proves that MKL can not be used for feature selection.

Weights β & Feature Selection. In this case, we used linear kernels with various number of features and applied MKL on them. The C was equal to 1 and 0.001 and normalized as well as normalized kernels have been compared. The weights obtained from MKL have been shown in Table. When un-normalized kernels are used, MKL assigns weight 1 to kernel with features=1000 while weight 0 to all other kernels. Comparing with individual kernel (Table 3.5), 1000 features kernel has the least classification accuracy; still it received the highest weight. With other combination of features as base kernel, MKL always provided highest weight to kernel learned from highest number of features. Even normalization did not change the weight distribution by much. 1000 feature kernel still received the highest weight.

Figure 3.1 Comparison of various models with increasing number of features (SNPs).



4 Conclusion

We have applied MKL to SNP genotype for type-1 diabetes risk prediction. We found that MKL has its limitations in providing better prediction accuracy than the base kernels. One of the caveats with MKL is that for solving the objective function, the base kernels have to be normalized. The normalization of the kernels brings all the data points on a unit scale, losing "information" that may be useful for classification. With un-normalized kernels, we have shown that MKL gives high weighage to kernel with highest dimenstions.

Other cavet of MKL is the choice of base kernels. MKL highly depends on the base kernels and choice of standard kernels is difficult to make. We tried standard kernels that are most often used in pattern recognition.

References

[1] Wray, Naomi R., Michael E. Goddard, and Peter M. Visscher. "Prediction of individual genetic risk to disease from genome-wide association studies." Genome research 17, no. 10 (2007): 1520-1528.

[2] Burton, Paul R., David G. Clayton, Lon R. Cardon, Nick Craddock, Panos Deloukas, Audrey Duncanson, Dominic P. Kwiatkowski et al. "Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls." Nature 447, no. 7145 (2007): 661-678.

[3] Yoon, Dankyu, Young J. Kim, and Taesung Park. "Phenotype prediction from genome-wide association studies: application to smoking behaviors." BMC systems biology 6, no. Suppl 2 (2012): S11.

[4] Wei, Zhi, Kai Wang, Hui-Qi Qu, Haitao Zhang, Jonathan Bradfield, Cecilia Kim, Edward Frackleton et al. "From disease association to risk assessment: an optimistic view from genome-wide association studies on type 1 diabetes." PLoS genetics 5, no. 10 (2009): e1000678.

[5] Mittag, Florian, Finja Büchel, Mohamad Saad, Andreas Jahn, Claudia Schulte, Zoltan Bochdanovits, Javier Simón-Sánchez et al. "Use of support vector machines for disease risk prediction in genome-wide association studies: Concerns and opportunities." Human Mutation 33, no. 12 (2012): 1708-1718.

[6] Do, Chuong B., David A. Hinds, Uta Francke, and Nicholas Eriksson. "Comparison of Family History and SNPs for Predicting Risk of Complex Disease." PLoS genetics 8, no. 10 (2012): e1002973.

[7] J.S. Bridle, "Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition," *Neurocomputing—Algorithms, Architectures and Applications*, F. Fogelman-Soulie and J. Herault, eds., NATO ASI Series F68, Berlin: Springer-Verlag, pp. 227-236, 1989. (Book style with paper title and editor)

[8] W.-K. Chen, *Linear Networks and Systems*. Belmont, Calif.: Wadsworth, pp. 123-135, 1993. (Book style)

[9] H. Poor, "A Hypertext History of Multiuser Dimensions," *MUD History*, http://www.ccs.neu.edu/home/pb/mud-history.html. 1986. (URL link *include year)

[10] K. Elissa, “An Overview of Decision Theory," unpublished. (Unplublished manuscript)

[11] R. Nicole, "The Last Word on Decision Theory," *J. Computer Vision,* submitted for publication. (Pending publication)

[12] C. J. Kaufman, Rocky Mountain Research Laboratories, Boulder, Colo., personal communication, 1992. (Personal communication)

[13] D.S. Coming and O.G. Staadt, "Velocity-Aligned Discrete Oriented Polytopes for Dynamic Collision Detection," *IEEE Trans. Visualization and Computer Graphics*, vol. 14, no. 1, pp. 1-12, Jan/Feb 2008, doi:10.1109/TVCG.2007.70405. (IEEE Transactions)

[14] S.P. Bingulac, “On the Compatibility of Adaptive Controllers,” *Proc. Fourth Ann. Allerton Conf. Circuits and Systems Theory*, pp. 8-16, 1994. (Conference proceedings)

First A. Author Biographies should be limited to one paragraph consisting of the following: sequentially ordered list of degrees, including years achieved; sequentially ordered places of employ concluding with current employment; association with any official journals or conferences; major professional and/or academic achievements, i.e., best paper awards, research grants, etc.; any publication information (number of papers and titles of books published); current research interests; association with any professional associations.

Second B. Author Jr. biography appears here. Degrees achieved followed by current employment are listed, plus any major academic achievements.

Third C. Author is a member of the IEEE and the IEEE Computer Society.

Table 3.1 **MKL (K=1) vs SVMlight**. Comarpison of MKL (one base kernel) and SVMlight at C=1,0.001 for 20, 40, 60, 80…1000 SNPs

Method	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 3.2 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%

Table 3.3 Comparison MKL with Linear, Polynomial and Gaussian (RBF) as Base Kernel to Linear Normalized Kernel

$K_{MKL} = \beta_1 K_{Linear} + \beta_2 K_{polyd2} + \beta_3 K_{polyd3} + \beta_4 K_{RBF1.2} + \beta_5 K_{RBF2} + \beta_6 K_{RBF5}$

Model/SNPs	20	40	60	80	100	200	400	600	800	1000
MKL	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

Table 3.4 Comparison of MKL with various number features to individual linear kernels (Normalized & un-normalized).

$K_{MKL} = \beta_1 K_{20} + \beta_2 K_{40} + \beta_3 K_{60} + \beta_4 K_{80} + \beta_5 K_{100} + \beta_6 K_{200} + \beta_7 K_{400} + \beta_8 K_{600} + \beta_9 K_{800} + \beta_{10} K_{1000}$

Model	C	20	40	60	80	100	200	400	600	800	1000	MKL
Linear Unnormal-ized	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

Table 3.5 Weights β from MKL with Various Models as Base Kernel

$K_{MKL} = \beta_1 K_{Linear} + \beta_2 K_{polyd2} + \beta_3 K_{polyd3} + \beta_4 K_{RBF1.2} + \beta_5 K_{RBF2} + \beta_6 K_{RBF5}$

SNPs/Models	β_1 Linear	β_2 Poly (d=2)	β_3 Poly (d=3)	β_4 RBF (γ=1.2)	β_5 RBF (γ=2)	β_6 RBF (γ=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 3.6 Weights β from MKL with various features (SNP) as base kernel
$K_{MKL} = \beta_1 K_{20} + \beta_2 K_{40} + \beta_3 K_{60} + \beta_4 K_{80} + \beta_5 K_{100} + \beta_6 K_{200} + \beta_7 K_{400} + \beta_8 K_{600} + \beta_9 K_{800} + \beta_{10} K_{1000}$

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

Figure 3.2 a) Comparision of MKL and linear kernel with C=1

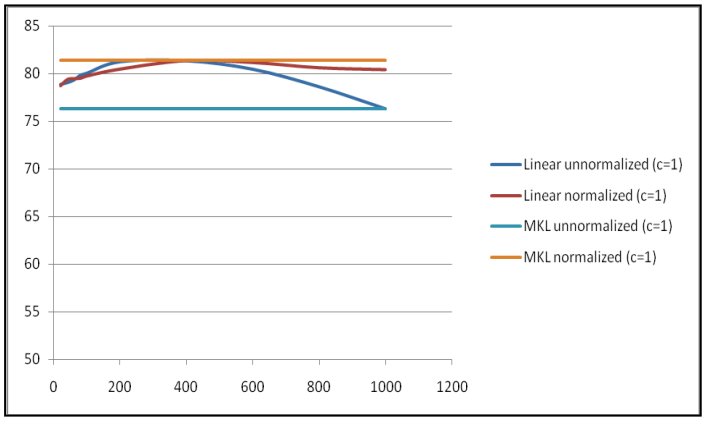


Figure 3.2 b) Comparision of MKL and linear kernel with C=0.001