Improved Genomic Selection using Vowpal Wabbit with Random Fourier Features

Jiaqin Jaslyn Zhang

A thesis submitted to the Department of Statistics for honors
Duke University
Durham NC 27705, USA
2017

Abstract. Nonlinear regression models are often used in statistics and machine learning due to greater accuracy than linear models. In this work, we present a novel modeling framework that is both computationally efficient for high-dimensional datasets, and predicts more accurately than most of the classic state-of-the-art predictive models. Here, we couple a nonlinear random Fourier feature data transformation with an intrinsically fast learning algorithm called Vowpal Wabbit or VW. The key idea we develop is that by introducing nonlinear structure to an otherwise linear framework, we are able to consider all possible higher-order interactions between entries in a string. The utility of our nonlinear VW extension is examined, in some detail, under an important problem in statistical genetics: genomic selection (i.e. the prediction of phenotype from genotype). We illustrate the benefits of our method and its robustness to underlying genetic architecture on a real dataset, which includes 129 quantitative heterogeneous stock mice traits from the Wellcome Trust Centre for Human Genetics.

Keywords: Fourier Transforms, Vowpal Wabbit, Genomic Selection

1 Introduction

The underlying problem in genomic selection is to explain variation in the phenotype from variation in the genotype across the entire genome [25]. This approach allows handling of complex traits that are regulated by many genes, each exhibiting a small effect [26] and has been applied to predict complex traits in human [66], animal [26], and plant populations [32, 33, 69]. However, implementing genomic selection in reality is often difficult due to the curse of dimensionality [63] because the number of gene markers, p, is much greater than the number of observations in the sample, n [21]. For instance, developed in the 1950s [28,29], BLUP (best linear unbiased prediction) has gained popularity in animal and plant breeding [23–25, 55, 58] as well as among human genetics [16, 43, 58, 67]. BLUP and its variants (G-BLUP, SNP-BLUP and H-BLUP) use a $n \times p$ genomic similarity matrix [35] to encode genetic similarities between pairs of individuals [58]. This matrix can become very large and increase computing complexity as the number of samples and markers grows [22].

Another challenge in genomic selection is identifying epistasis. Epistasis is defined as the interaction between multiple genes [64] and has long been recognized as an important component in dissecting genetic pathways and understanding the evolution of complex genetic systems [33, 50]. Modeling epistasis increases phenotype prediction accuracy [12, 33, 46] and explains missing heritability - the proportion of heritability not explained by the top associated variants from Genome-wide association studies [12,18,61]. Thus, many statistical methods explicitly search for pairwise or higher-order interactions to model epistasis [12]. However, the extremely large search space (e.g. p(p+1)/2 pairwise combinations for p variants) poses heavy computational burden to these methods and limits their statistical power. A recent development is the Bayesian approximate kernel regression model (BAKR) [11]. By introducing a general effect size for every

marker that represents both its additive effects and its combined epistatic effects with all other variants, BAKR avoids the need to explicitly observe all possible interactions, thus achieving some moderate computational efficiency. However, because of its Bayesian framework and its dependency on inference via posterior sampling, analyses with this model still takes hours or even days to run on particularly large datasets [11].

Our main contribution in this paper is the development of a nonlinear regression framework that predicts as accurately as state-of-the-art methods in machine learning and statistics, but with greater computational efficiency. The key idea is based on adding random Fourier features to an otherwise linear intrinsically fast learning algorithm Vowpal Wabbit (VW). This newly modified VW framework with nonlinear structure not only maintains the computational speed of the original software, but now also considers all possible higher-order interactions between markers, thus achieving better predictive accuracy.

In Section 2, we review the main learning algorithm underlying the VW framework, and detail how we introduce random Fourier features into the model. In Section 3, we examine the predictive performance of our proposed framework, as opposed to other comparable methods, on real data. In Section 4, we emphasize the importance of the inclusion of non-linearity by comparing results from linear and nonlinear VW. We close with a discussion for future work.

2 Nonlinear Vowpal Wabbit

In this section, we detail our extension of the Vowpal Wabbit (VW) framework with random Fourier features. The intuition behind the proposed utility of this model is based on the following three observations: (1) VW is an intrinsically fast linear learning algorithm [5]; (2) in high-dimensional regression, smooth nonlinear functions are more predictive than linear functions [11]; (3) the inclusion of random Fourier features can preserve the advantages of both linear and nonlinear modeling approaches. In the remainder of this section, we introduce the online learning system Vowpal Wabbit and its main algorithm.

2.1 Learning Algorithm

Vowpal Wabbit (VW) is a learning system sponsored by Microsoft Research and (previously) Yahoo! Research [38]. More specifically, it is a hybrid of both the stochastic gradient descent and the batch learning algorithms. Briefly, in stochastic gradient descent, an input is read in in a sequential order and is used to update predictors for future data at each step. Conversely, batch learning techniques calculate predictors by learning on the entire data set at once [2]. The basic learning algorithm of VW is as follows:

Algorithm 1 Updating rules for each sample with squared loss function

1: Get features of current sample $\mathbf{x}$
2: Get most updated weights $\mathbf{w}$
3: Make prediction $\hat{y} \leftarrow \sum_i w_i x_i$
4: Learn truth y with importance I
5: Update $w_i \leftarrow w_i + 2\eta(y - \hat{y})I$

Here, $\mathbf{x}$ is a vector of features, and $\mathbf{w}$ is a vector of all the most updated weights for those features. The weights by default are initially set to be 0 for all features at the first iteration. A separate parameter η is used to specify how aggressively the algorithm steps in a negative direction from the gradient of the initial guess. For each algorithmic step, η is updated as follows:

$$\eta_t = \lambda d^k \left(\frac{t_0}{t_0 + w_t} \right)^p, \tag{1}$$

where w_t is the sum of the importance weights for all samples seen up to step t, λ and t_0 are searched in a logarithmic scale, d represents the decay learning rate between passes, and p specifies the power on the learning rate decay (i.e. $p = 0$ means the learning rate does not decay) [36].

In a single pass, VW repeats the above procedures for each available sample and keeps updating the weights until it reaches the end of the training data.

2.2 Combining Random Fourier Features in Vowpal Wabbit

Despite its computational efficiency, Vowpal Wabbit is not the most desirable framework for genomic selection. It is essentially a linear regression and, in the presence of high-dimensional sequencing data, it becomes infeasible to explicitly include all possible higher-order interactions between genetic variants in a fast and principled way. Therefore, we propose using random Fourier features in combination with VW for higher prediction accuracy. To do this, we will use certain function analytic properties of a specific type of an infinite dimensional function space, called a reproducing kernel Hilbert space (RKHS), to develop this regression framework. We will consider the RKHS to be defined by a particular type of kernel function called a Mercer kernel [44].

2.2.1 Reproducing kernel Hilbert Spaces Consider the following regression model:

$$y_i = f(\mathbf{x}_i) + \epsilon_i, \quad \epsilon_i \overset{iid}{\sim} N(0, \sigma^2), \quad f \in H \tag{2}$$

where we are given n observations $\{\mathbf{x}_i, y_i\}_{i=1}^n$ of covariates $\mathbf{x}_i \in \mathbb{R}^p$ and responses $y_i \in \mathbb{R}$. One can define a RKHS starting with a positive definite kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. The eigenfunctions $\{\psi_j\}_{j=1}^\infty$ and eigenvalues $\{\lambda_j\}_{j=1}^\infty$ of the following integral operator defined by the kernel function:

$$\lambda_j \psi_j(\mathbf{u}) = \int_x k(\mathbf{u}, \mathbf{v}) \psi_j(\mathbf{v}) d\mathbf{v}, \tag{3}$$

can be used to define the RKHS. For a Mercer kernel [44] the following expansion holds

$$k(\mathbf{u}, \mathbf{v}) = \sum_{j=1}^{\infty} \lambda_i \langle \Psi_j(\mathbf{u}), \Psi_j(\mathbf{v}) \rangle, \tag{4}$$

and the RKHS can be formally defined as a linear combination of the bases $\{\psi_j\}_{j=1}^{\infty}$ such as

$$\mathcal{H} = \left\{ f | f(\mathbf{x}) = \sum_{j=1}^{\infty} c_j \psi_j(\mathbf{x}), \forall \mathbf{x} \in \mathcal{X} \text{ and } \|f\|_{\mathrm{K}} \leq \infty \text{ with } \|f\|_{\mathrm{K}}^2 = \sum_{j=1}^{\infty} \frac{c_j^2}{\lambda_j^2} \right\} \tag{5}$$

here $\|f\|_{\mathrm{K}}$ is the RKHS norm. We will define $\boldsymbol{\psi}(\mathbf{x})$ as a vector called the feature space, with basis elements $\{\sqrt{\lambda_i}\psi_i(\mathbf{x})\}_{i=1}^{\infty}$. We can also specify coefficients $\mathbf{c}$, a vector with elements $\{\mathrm{c}_i\}_{i=1}^{\infty}$, in the feature space. The RKHS can now be defined as the following:

$$\mathcal{H}_{\mathrm{K}} = \left\{ f \,|\, f(\mathbf{x}) = \boldsymbol{\psi}(\mathbf{x})^{\intercal}\mathbf{c}, \, \forall \mathbf{x} \in \mathcal{X} \text{ and } \|\mathbf{c}\|_{\mathrm{K}}^2 < \infty \right\}.$$

Note that the above specification of an RKHS looks very much like a linear regression model except the bases are $\boldsymbol{\psi}(\mathbf{x})$ rather than the unit basis, and the space can be infinite-dimensional. While this bases representation exists in theory, it is extremely hard to model in practice.

2.2.2 The Kernel Trick Using the Representer theorem [3, 11, 51, 53], one can approximate $f(\mathbf{x})$ as:

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}, \mathbf{x}_i).$$

This representation is often used in kernel models or Gaussian processes where (in the context of genetics and genomics) $\mathbf{x}_i$ is a vector of genotypes, and the kernel function $k(\mathbf{u}, \mathbf{v})$ is a similarity measure. The resulting kernel matrix takes the following form:

$$\mathbf{K} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \dots & k(\mathbf{x}_1, \mathbf{x}_n) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \dots & k(\mathbf{x}_2, \mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & k(\mathbf{x}_n, \mathbf{x}_2) & \dots & k(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix}$$

This "kernel trick" reduces the ∞-dimensional problem to a n-dimensional optimization problem with n parameters. As a result, we can specify the subspace of the RKHS that realized by the data as:

$$\mathcal{H}_{\mathrm{X}} = \left\{ f | f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}, \mathbf{x}_i), \{\alpha_i\} \in \mathbb{R}^n, \|f\|_K^2 < \infty \right\}, \tag{6}$$

However, as shown above, using the full kernel matrix may become computationally expensive when scaling to large size datasets. A dataset with half a million training samples might take days to train [3, 51].

2.2.3 Random Fourier Features We now specify a methodology used to directly map from a kernel space to a relatively low-dimensional predictor space using a randomized feature map [3, 11, 51].

The main idea that we detail here is based on Bochner's theorem, which relates positive definite shift-invariant kernel function to their Fourier transforms [3, 40, 41, 51, 53]. Hence, we will restrict our focus to kernel functions that are shift-invariant and integrate to one. That is: $k(\mathbf{u}, \mathbf{v}) = k(\mathbf{u}-\mathbf{v})$ and $\int k(\mathbf{z})d\mathbf{z} = 1$, with $\mathbf{z} = \mathbf{u} - \mathbf{v}$ [11]. Then, according to Bochner's Theorem [4] the following expansion holds:

$$k(\|\mathbf{x}_i - \mathbf{x}_j\|) = \int_{\mathbb{R}^p} f(\boldsymbol{\omega}) \exp\{\iota\boldsymbol{\omega}^\intercal(\mathbf{x}_i - \mathbf{x}_j)\}d\boldsymbol{\omega} = \mathbb{E}_{\boldsymbol{\omega}}[\eta_{\boldsymbol{\omega}}(\mathbf{x}_i)\eta_{\boldsymbol{\omega}}(\mathbf{x}_j)^*] \quad (7)$$

where $\eta_{\boldsymbol{\omega}}(\mathbf{x}_i) = \exp(\iota\boldsymbol{\omega}^\intercal\mathbf{x}_i)$, and the Fourier transform of the kernel function $f(\mathbf{w})$ is formally defined as [3, 11, 51]:

$$f(\boldsymbol{\omega}) = \int_X k(\mathbf{x})e^{-\iota 2\pi\boldsymbol{\omega}^\intercal\mathbf{x}}d\mathbf{x}. \quad (8)$$

Finally, we follow previous works [51], and use the following procedure for generating random Fourier features from our original data:

Algorithm 2 Random Fourier Features.

1: **Require:** A positive definite shift-invariant kernel that integrates to one: $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x} - \mathbf{x}')$ *and* $\int k(\mathbf{x} - \mathbf{x}')d(\mathbf{x} - \mathbf{x}') = 1$.
2: **Ensure:** A randomized feature map $\tilde{\boldsymbol{\psi}}(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^D$so that $\tilde{\boldsymbol{\psi}}(\mathbf{x})^\intercal\tilde{\boldsymbol{\psi}}(\mathbf{x}') \approx k(\mathbf{x} - \mathbf{x}')$.
3: Compute the Fourier transform of the kernel $k : f(\omega) = \frac{1}{2\pi}\int e^{-\iota\omega^\intercal\Delta}k(\Delta)d\Delta$ with $\Delta = \mathbf{x} - \mathbf{x}'$.
4: Draw $\boldsymbol{\omega}_k \overset{iid}{\sim} f(\boldsymbol{\omega})$ for $k = 1, \ldots, D$.
5: Let $\tilde{\boldsymbol{\psi}}(\mathbf{x}) \equiv \sqrt{\frac{2}{D}}[\cos(\boldsymbol{\omega}_1\mathbf{x}_1), \sin(\boldsymbol{\omega}_1\mathbf{x}_1), \ldots, \cos(\boldsymbol{\omega}_D\mathbf{x}_D), \sin(\boldsymbol{\omega}_D\mathbf{x}_D)]^\intercal$.

We now may rewrite the VW learning algorithm to include the nonlinear random Fourier features:

Algorithm 3 Updating rules for the Nonlinear VW

1: Get random Fourier feature $\tilde{\boldsymbol{\psi}}(\mathbf{x})$
2: Get most updated weights $\mathbf{w}$
3: Make prediction $\hat{y} \leftarrow \sum_i w_i\tilde{\psi}(x_i)$
4: Learn truth y with importance I
5: Update $w_i \leftarrow w_i + 2\eta(y - \hat{y})I$

2.2.4 Feature Hashing Recall that due to the potentially large sample sizes in genome-wide association studies (GWASs) [6, 9, 47, 52], utilizing the entire kernel matrix can occupy a large amount of computing memory. VW works around this issue by replacing the full kernel matrix with a "hash kernel" matrix. The formal definition of a hash kernel is as follows [37, 56, 65]:

Definition 1. *Denote h to be a hash function such that $h : \mathbb{N} \to \{1, \ldots, m\}$. Moreover, denote ξ to be a hash function such that $\xi : \mathbb{N} \to \{\pm 1\}$. Then for vectors $\boldsymbol{x}, \boldsymbol{x}' \in L^2$, we define a hashed feature map ϕ and the corresponding inner product*

$$\phi_i^{(h,\xi)}(\boldsymbol{x}) = \sum_{j:h(j)=i} \xi(j)\boldsymbol{x}_j \tag{9}$$

$$\langle \boldsymbol{x}, \boldsymbol{x}' \rangle_\phi := \langle \phi^{(h,\xi)}(\boldsymbol{x}), \phi^{(h,\xi)}(\boldsymbol{x}') \rangle \tag{10}$$

The feature hashing strategy contributes to VW's computational efficiency by hashing high dimensional input vectors into a lower dimensional feature space [39, 56]. The input space is therefore reduced from $\mathbb{R}^n$ to $\mathbb{R}^m$, where m $\ll$ n.

2.3 Comparable Methods

In this work, we compare the predictive performance of the nonlinear VW framework with five other state-of-art genomic selection methods as well as the regular linear VW. We provide a brief description of these competing methods here:

1. **Bayesian Ridge Regression (BRR).** A Bayesian hierarchical model which assumes that the effect size β_j for each j^{th} variant comes from the following normal distribution [30]:

   $$\beta_j \sim \text{N}(0, \sigma^2) \text{with} \quad \sigma^2 \sim \text{Scaled-Inv-}\chi^2(\nu, \phi)$$

   where N represents a normal distribution with 0 mean and variance σ^2, and Scaled-Inv-χ^2 denotes a scaled-inverse chi-squared distribution with degrees of freedom and scale hyper-parameters ν and ϕ, respectively. Following previous studies, we set $\nu = 5$ and $\phi = 2/5$ [17]. We use the R package BGLR [1, 14, 15, 17, 30] to sample from the posterior distribution of $\boldsymbol{\beta}$.
2. **Bayesian Lasso (BL).** A model which assumes that the effect size β_j for each j^{th} variant follows a double-exponential or Laplacian prior [42, 48, 49]:

   $$\beta_j \sim \text{DE}(0, \tau^{-1}) \quad \text{with} \quad \tau^2 \sim \text{Ga}(\kappa_1, \kappa_2),$$

   where DE denotes the double exponential (Laplace) distribution with mean 0 and scale parameter τ^{-1}, and Ga denotes a Gamma distribution with shape and rate parameters κ_1 and κ_2, respectively. Following previous studies [48], we allow τ^2 to follow a conjugate Gamma prior, and set $\kappa_1 = 0.55$ and $\kappa_2 = 10^{-6}$ — this is consistent with other statistical genetics simulations [17, 70]. We again use the R package BGLR to sample from the posterior distribution of $\boldsymbol{\beta}$.

3. **Bayesian Linear Mixed Model (BLMM).** An extension of the regular linear model that also includes random effects to control for population structure. For this paper, we assume that the random effects are normally distributed with mean vector $\mathbf{0}$ and covariance matrix $\mathbf{K}=\mathbf{X}\mathbf{X}^{\intercal}/p$, where $\mathbf{K}$ is commonly referred to as a linear or additive kernel matrix [11, 33, 34].
4. **Support Vector Machine (SVM).** A member of the kernel prediction and classification methodology family, which uses a supervised RKHS learning algorithm. In this work, we use the ksvm function in the kernlab R package under the rbfdot model setting (i.e. Gaussian radial basis function). Note that the SVM estimates its parameters deterministically and does not require MCMC [7, 8, 10, 11, 19, 31, 34, 54, 57, 62, 68].
5. **Bayesian Approximate Kernel Regression (BAKR).** A nonlinear Bayesian kernel regression framework that provides an analogue of the effect size for each explanatory variable, particularly when the kernel is shift-invariant [11]. Specifically, for this paper, we choose to approximate a Gaussian kernel with band-width parameter $h = 1$, utilize an empirical factor representation in which we keep eigenvectors explaining 80% of the cumulative variance in eigenvalues, and set model hyper-parameters $\nu = 5$ and $\phi = 2/5$.

For all MCMC based methods, we run 50,000 iterations with the first 25,000 iterations being used as a burn-in. We note that longer MCMC chains neither greatly improved, nor markedly hindered, the association mapping performance of any of the previously mentioned methods.

3 Results

We assess the prediction accuracy of our model, and the aforementioned comparative models, by analyzing a heterogeneous stock mouse dataset from the Wellcome Trust Centre for Human Genetics [59] (`http://mus.well.ox.ac.uk/mouse/HS/`). The dataset contains $n \approx 2,000$ individuals and $p \approx 10,000$ SNPs (exact numbers varying slightly depending on the phenotype). This is the ideal dataset for assessing predictive performance, not only because it contains a wide variety of quantitative phenotypes, but also because the observed mice are related. Relatedness has been suggested to manifest different orders of interaction effects [27,71]. Lastly, note that each phenotype is standardized before being analyzed. For each phenotype, we randomly set aside 20% of the data for testing, and use the rest 80% as the training set. We repeat this procedure 10 times, and assess model accuracy using mean square prediction error (MSPE).

3.1 Vowpal Wabbit Software Parameters

The VW software provides two learning modes: (1) simultaneous learning and prediction; (2) prediction post model learning. We use the latter mode for the current application in order to keep comparisons consistent with the other methods considered. Given the high-dimensional nature of the dataset, we also allow

VW to cycle over the training set 100 times in order to ensure it is exhibiting its optimal performance.

We run Vowpal Wabbit (VW) on the original mouse dataset, as well as on three datasets that have been generated after applying the random Fourier feature transformation to approximate a Gaussian kernel. Here, we consider three different values of total random draws D: 1000, 2000 and 5000 respectively. Note that these choices of values are rather arbitrary, and ideally we would like to select this parameter in a more principled manner (e.g. objective Bayesian sampling). This is a direction we will explore in future work.

It is also useful to note that the Vowpal Wabbit software provides numerous command line arguments. Due to time constraints, we were not able to fully explore all of their utilities. Nevertheless, we identified what we believed to be the most relevant parameters and attempted to optimize model performance over those. The final values of these arguments are summarized in Table 1. For a full list of command line arguments, we refer to the following software website (`https://github.com/JohnLangford/vowpal_wabbit/`).

Specifically, we performed a grid search on 9 different learning rates (ranging from 0.01 to 0.09) to ensure the best results for each of the 4 versions of VW (i.e. the original dataset, as well as the 3 datasets after random Fourier feature transformation with different D values). Figure 1 summarizes the prediction accuracy for different learning rates under each of the four settings averaged over all of the 129 quantitative mice phenotypes. It is clear that average MSPE increases as learning rate increases. Learning rate of 0.01 consistently gives the minimum average MSPE in all settings. This observation is reasonable since a higher learning rate will make the model converge faster, but it may over-fit and give worse predictions. From this point forward, we will only compare the performance of settings with the most ideal learning rate at 0.01 with other state-of-art methods.

Table 2 shows the average MSPE for random Fourier feature transformations with different D values across the 129 quantitative mice phenotypes. It also includes a metric Pr[Optimal], which is the percentage of the time that a setting exhibits the lowest MSPE. Values in bold mark the method with the lowest average MSPE and highest Pr[Optimal], respectively. The random Fourier feature transformation with $D = 5000$ not only gives the lowest average MSPE, but also most frequently performs best. It is not surprising that we choose this value of D since, after the random Fourier feature transformation, we get a n $\times$ 2D feature matrix. We assume that more features will equate in a better approximation of the infinite dimensional bases space. From this point forward, we will set $D = 5000$.

3.2 Model Comparison

We now compare our nonlinear VW framework with six other state-of-art genomic selection methods: the linear version of VW, the Bayesian Lasso, Bayesian ridge regression, Bayesian linear mixed model, support vector machine(SVM) and the Bayesian approximate kernel regression model.

Figure 2 summarizes the predictive performance for each phenotype for Bayes Ridge, Bayes Lasso, Bays LMM, SVM, BAKR, linear VW, and VW with random Fourier features, as assessed by measuring MSPE based on ten fold cross-validation. Note the VW model with random Fourier features is marked by a solid square.

Bayes Lasso, Bayes Ridge, Bayes LMM and linear VW had an average MSPE of 1.00, 1.01, 1.04 and 1.01, respectively. VW with random Fourier features performs better with an average MSPE of 0.98. This proves our assumption that adding non-linear features improves accuracy for genomic selection. Since our dataset includes related samples, VW with random Fourier features can take into account different orders of interaction effects, and thus improve prediction accuracy. However, BAKR and SVM outperformed our framework with an average MSPE of 0.87 and 0.88, respectively. Since VW has around 100 software parameters, the performance of the model may heavily depend on optimizing over all of these parameters. In this paper, we only did a grid search on learning rates and the number of random Fourier features; however, we simply used the default setting for most other parameters. An optimization on all parameters should greatly improve this predictive performance.

We want to point that while this particular dataset is smaller than most GWASs, it can become computationally demanding to analyze much larger studies with BAKR. In particular, the computational complexity of VW scales linearly with the number of genetic markers and samples. Likewise, the complexity of the SVM is linear on the number of the support vectors and linear on the number of features. Due to the inverse projection BAKR employs in order to do inference on covariates in the original feature space, BAKR must use an inverse which is cubic in complexity.

3.3 Variance Component Analysis

In order to explain why the nonlinear models outperformed the parametric models in nearly all of the 129 phenotypes, we used a variance component analysis to evaluate the overall contribution of nonlinearities to the phenotypic variance explained, or PVE (see [11, 12] for details). The basic idea for computing PVE is using a linear mixed model with multiple variance components to partition the phenotypic variance into four different components: a linear component, a pairwise interaction component, a third order interaction component, and a common environmental component shared by mice within the same cage. Disregarding any random noise, we quantify the contribution of the four components by examining the portion of PVE (pPVE) explained by that component.

Figure 3 displays the PVE decomposition in this mouse dataset. The significant finding in this plot, is that very rarely are the linear and pairwise interaction components combined for the greatest contributors in explaining variation in each response. This variance component analysis highlights the importance of accounting for modeling interaction effects when carrying out genomic selection. We again stress that the advantage of these nonlinear models lies in the

explicit modeling of interaction relationships between covariates and the desired response.

4 Discussion

In this paper, we have presented a model that is both computationally efficient for high-dimensional datasets, and predicts more accurately than most of your classic state-of-the-art linear predictive models. We combined a nonlinear random Fourier feature data transformation with an intrinsically fast online learning algorithm called Vowpal Wabbit (VW). By doing so, our framework exhibits all the efficient methods of the original VW framework, which helps accelerate the computing process and save memory space. At the same time, introducing nonlinear structure to an essentially linear algorithm boosts its prediction accuracy for genomic selection. We tested the utility of our VW extension to a real dataset, which included 129 quantitative heterogeneous stock mice traits collected by the Wellcome Trust Centre for Human Genetics. This dataset had $\sim$ 2000 mice that were genotyped at $\sim$ 10,000 SNPs. The nonlinear extension of the VW model predicts better than the regular VW, BL, Bayes Ridge, and BLMM. Despite the fact that it failed to outperform BAKR or the SVM, this novel framework still runs much faster than all the other methods.

Our model is not without its limitations. For this study, we neglected the fact that some mice may come from different families and live in different cages and environment's potential impact on phenotypes [60]. In the future, we want to partition the dataset to handle these potential cage effects. Currently, we only performed a grid search for the software learning rates over a relatively small interval ranging from 0.01 to 0.09. In addition, we did not tune all the relevant command line parameters of VW in a data driven way. For future work, it would be nice to perform Bayesian Optimization over these choices by placing priors on the tuning parameter space and learning them with some uncertainty — the caveat being that such an optimization approach might take away from the speed of VW. It would be necessary to come up with an alternative to learn the tuning parameters effectively without compromising the most notable features of VW.

Given the flexibility of Vowpal Wabbit and random Fourier feature transformation, our framework can work with any kernel function that is shift-invariant. Therefore, one important direction of future research would be exploring the use of our framework with different kernels that may improve prediction accuracy while preserving the speed. Possible kernel choices include the popular Matérn covariance function or the rational quadratic covariance function in spatial statistics and machine learning literatures [11, 13, 20, 45]. Furthermore, although the model we have developed in this paper can be used in many fields, we have only focused on applying it to a single dataset in a single area of genomic selection in this study. It would be interesting to see how our framework performs on other gene related datasets and outside the genomic selection field.

5 Software

Software for implementing the random Fourier transformations used in this paper is mainly carried out in C++ code, which is available to the public at `https://github.com/lorinanthony/BAKR`. The Vowpal Wabbit algorithm is also publicly available at `https://github.com/JohnLangford/vowpal_wabbit`.

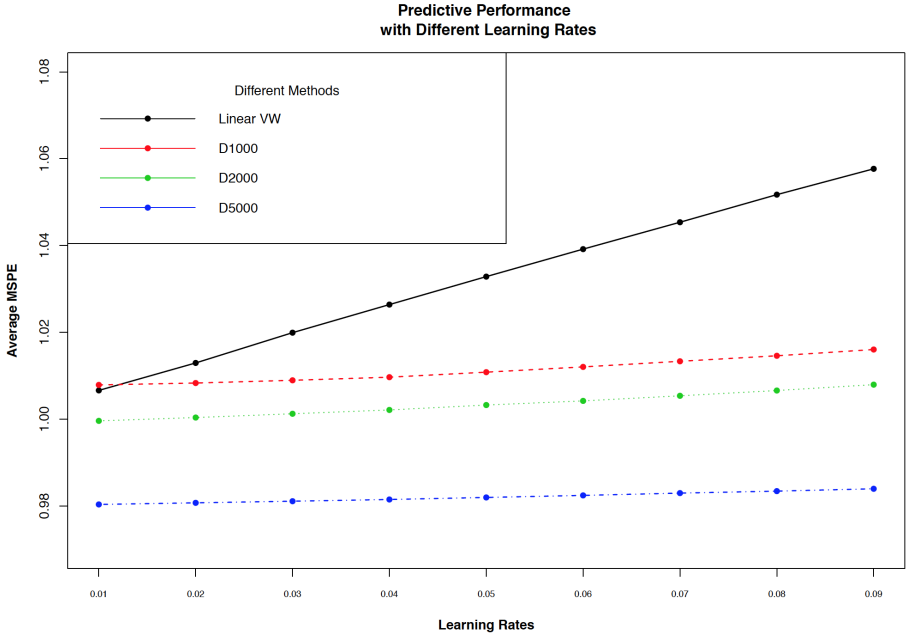
6 Acknowledgements

JZ would like to thank Lorin Crawford and Sayan Mukherjee for useful conversations and suggestions on previous versions of the manuscript.

Parameter	Final Values	Usage
l	0.01	Learning Rate
passes	100	Number of Training Passes
loss function	Squared loss	Specify the loss function to be used

Table 1. Relevant command line arguments

Fig. 1. Comparisons of Average MSPE for Different Learning Rates



D Values	Average MSPE	Standard Deviation	Pr[Optimal]
5000	**0.98**	0.09	**0.44**
2000	1.00	0.12	0.30
1000	1.01	0.11	0.26

Table 2. Average MSPE and Pr[Optimal] for Different D Values

Fig. 2. Predictive Performance of VW with RFF and other state-of-art methods

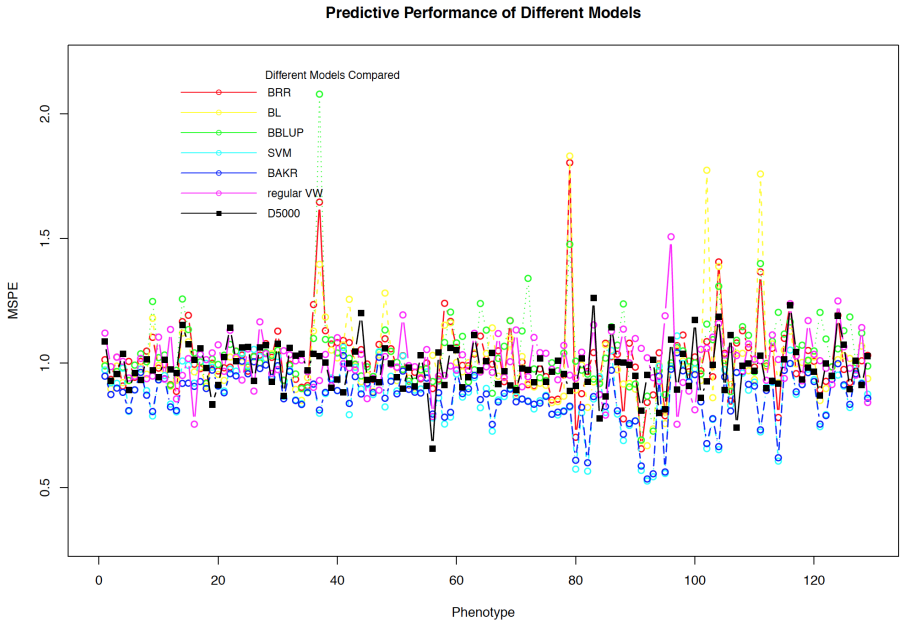
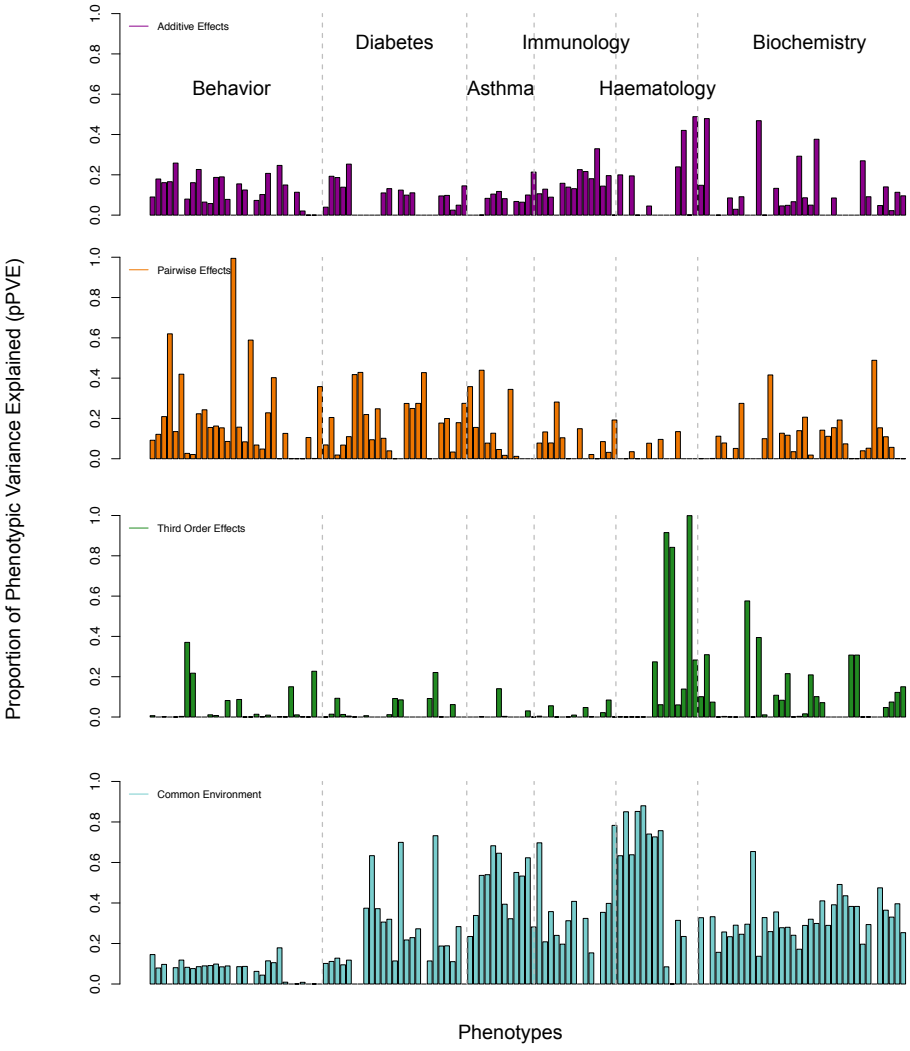


Fig. 3. The portion of PVE (pPVE) calculated by the estimated variance components, each of which represent a particular genetic effect of interest. This figure was first published in [11].



References

1. Cran - package bglr. `https://cran.r-project.org/web/packages/BGLR/index.html`. (Accessed on 03/31/2017).
2. A. Agarwal, O. Chapelle, M. Dudík, and J. Langford. A reliable effective terascale linear learning system. *Journal of Machine Learning Research*, 15(1):1111–1133, 2014.
3. E. G. Băzăvan, F. Li, and C. Sminchisescu. Fourier kernel learning. In *Computer Vision–ECCV 2012*, pages 459–473. Springer, 2012.
4. S. Bochner. A theorem on Fourier-Stieltjes integrals. *Bulletin of the American Mathematical Society*, 40:271–276, 1934.
5. L. Bottou. projects:sgd [leon.bottou.org]. `http://leon.bottou.org/projects/sgd`. (Accessed on 03/31/2017).
6. P. R. Burton, D. G. Clayton, L. R. Cardon, N. Craddock, P. Deloukas, A. Duncanson, D. P. Kwiatkowski, M. I. McCarthy, W. H. Ouwehand, N. J. Samani, et al. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature*, 447(7145):661–678, 2007.
7. O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine learning*, 46(1-3):131–159, 2002.
8. V. Cherkassky and Y. Ma. Practical selection of svm parameters and noise estimation for svm regression. *Neural networks*, 17(1):113–126, 2004.
9. G. L. G. Consortium. Discovery and refinement of loci associated with lipid levels. *Nat Genet*, 45(11):1274–1283, 11 2013.
10. C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
11. L. Crawford, K. C. Wood, X. Zhou, and S. Mukherjee. Bayesian approximate kernel regression with variable selection. *arXiv preprint arXiv:1508.01217*, 2016.
12. L. Crawford, P. Zeng, S. Mukherjee, and X. Zhou. Detecting epistasis with the marginal epistasis test in genetic mapping studies of quantitative traits. *bioRxiv*, page 066985, 2017.
13. N. Cressie and H.-C. Huang. Classes of nonseparable, spatio-temporal stationary covariance functions. *Journal of the American Statistical Association*, 94(448):1330–1339, 1999.
14. J. Crossa, P. Pérez, J. Hickey, J. Burgueño, L. Ornella, J. Cerón-Rojas, X. Zhang, S. Dreisigacker, R. Babu, Y. Li, D. Bonnett, and K. Mathews. Genomic prediction in CIMMYT maize and wheat breeding programs. *Heredity*, 112(1):48–60, 01 2014.
15. G. de los Campos, D. Gianola, G. J. M. Rosa, K. A. Weigel, and J. Crossa. Semi-parametric genomic-enabled prediction of genetic values using reproducing kernel Hilbert spaces methods. *Genetics Research (Cambridge)*, 92(4):295–308, 08 2010.
16. G. de los Campos, J. M. Hickey, R. Pong-Wong, H. D. Daetwyler, and M. P. Calus. Whole-genome regression and prediction methods applied to plant and animal breeding. *Genetics*, 193(2):327–345, 2013.
17. G. de los Campos, H. Naya, D. Gianola, J. Crossa, A. Legarra, E. Manfredi, K. Weigel, and J. Cotes. Predicting quantitative traits with regression models for dense molecular markers and pedigree. *Genetics*, 182(1):375–385, 05 2009.
18. E. E. Eichler, J. Flint, G. Gibson, A. Kong, S. M. Leal, J. H. Moore, and J. H. Nadeau. Missing heritability and strategies for finding the underlying causes of complex disease. *Nature Reviews Genetics*, 11(6):446–450, 2010.
19. T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in computational mathematics*, 13(1):1, 2000.

20. M. G. Genton. Classes of kernels for machine learning: a statistics perspective. *Journal of machine learning research*, 2(Dec):299–312, 2001.
21. D. Gianola, G. Morota, and J. Crossa. Genome-enabled prediction of complex traits with kernel methods: What have we learned. In *Proceedings, 10th World Congress of Genetics Applied to Livestock Production*, page 6, 2014.
22. J. Gibson and J. Jeyaruban. The effects of blup evaluations, population size and restrictions on selection of close relatives on response and inbreeding in egg-laying poultry. *Proc. Nat. Breeders Roundtable, St. Louis, Missouri*, pages 1–7, 1993.
23. M. E. Goddard and B. Hayes. Genomic selection. *Journal of Animal breeding and Genetics*, 124(6):323–330, 2007.
24. D. Habier, R. L. Fernando, K. Kizilkaya, and D. J. Garrick. Extension of the bayesian alphabet for genomic selection. *BMC bioinformatics*, 12(1):186, 2011.
25. B. Hayes, M. Goddard, et al. Prediction of total genetic value using genome-wide dense marker maps. *Genetics*, 157(4):1819–1829, 2001.
26. B. J. Hayes, P. J. Bowman, A. Chamberlain, and M. Goddard. Invited review: Genomic selection in dairy cattle: Progress and challenges. *Journal of dairy science*, 92(2):433–443, 2009.
27. G. Hemani, S. Knott, and C. Haley. An evolutionary perspective on epistasis and the missing heritability. *PLoS Genet*, 9(2):e1003295, 2013.
28. C. R. Henderson. Estimation of genetic parameters. In *Biometrics*, volume 6, pages 186–187. INTERNATIONAL BIOMETRIC SOC 1441 I ST, NW, SUITE 700, WASHINGTON, DC 20005-2210, 1950.
29. C. R. Henderson, O. Kempthorne, S. R. Searle, and C. Von Krosigk. The estimation of environmental and genetic trends from records subject to culling. *Biometrics*, 15:192–218, 1959.
30. R. Howard, A. L. Carriquiry, and W. D. Beavis. Parametric and nonparametric statistical methods for genomic selection of traits with additive and epistatic genetic architectures. *G3 (Bethesda)*, 4(6):1027–1046, 06 2014.
31. R. Howard, A. L. Carriquiry, and W. D. Beavis. Parametric and nonparametric statistical methods for genomic selection of traits with additive and epistatic genetic architectures. *G3: Genes— Genomes— Genetics*, 4(6):1027–1046, 2014.
32. J.-L. Jannink, A. J. Lorenz, and H. Iwata. Genomic selection in plant breeding: from theory to practice. *Briefings in functional genomics*, 9(2):166–177, 2010.
33. Y. Jiang and J. C. Reif. Modeling epistasis in genomic selection. *Genetics*, 201(2):759–768, 2015.
34. S. S. Keerthi and C.-J. Lin. Asymptotic behaviors of support vector machines with gaussian kernel. *Neural computation*, 15(7):1667–1689, 2003.
35. M. Koivula, I. Strandén, G. Su, and E. A. Mäntysaari. Different methods to calculate genomic predictions—comparisons of blup at the single nucleotide polymorphism level (snp-blup), blup at the individual level (g-blup), and the one-step approach (h-blup). *Journal of dairy science*, 95(7):4065–4073, 2012.
36. J. Langford. Command line arguments · johnlangford/vowpal_wabbit wiki · github. `https://github.com/JohnLangford/vowpal_wabbit/wiki/Command-line-arguments`. (Accessed on 03/31/2017).
37. J. Langford. Feature hashing and extraction · johnlangford/vowpal_wabbit wiki · github. `https://github.com/JohnLangford/vowpal_wabbit/wiki/Feature-Hashing-and-Extraction`. (Accessed on 03/31/2017).
38. J. Langford. Home · johnlangford/vowpal_wabbit wiki · github. `https://github.com/JohnLangford/vowpal_wabbit/wiki`. (Accessed on 03/31/2017).
39. J. Langford, L. Li, and A. Strehl. Vowpal wabbit online learning project, 2007.

40. F. Li, Y.-S. Fu, Y.-H. Dai, C. Sminchisescu, and J. Wang. Kernel learning by unconstrained optimization. In *AISTATS*, pages 328–335, 2009.
41. F. Li, C. Ionescu, and C. Sminchisescu. Random fourier approximations for skewed multiplicative histogram kernels. In *Joint Pattern Recognition Symposium*, pages 262–271. Springer, 2010.
42. J. Li, K. Das, G. Fu, R. Li, and R. Wu. The Bayesian lasso for genome-wide association studies. *Bioinformatics*, 27(4):516–523, 02 2011.
43. R. Makowsky, N. M. Pajewski, Y. C. Klimentidis, A. I. Vazquez, C. W. Duarte, D. B. Allison, and G. de Los Campos. Beyond missing heritability: prediction of complex traits. *PLoS Genet*, 7(4):e1002051, 2011.
44. J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society, London A*, 209:415–446, 1909.
45. B. Minasny and A. B. McBratney. Spatial prediction of soil properties using eblup with the matérn covariance function. *Geoderma*, 140(4):324–336, 2007.
46. P. R. Muñoz, M. F. Resende, S. A. Gezan, M. D. V. Resende, G. de los Campos, M. Kirst, D. Huber, and G. F. Peter. Unraveling additive from nonadditive effects using genomic relationship matrices. *Genetics*, 198(4):1759–1768, 2014.
47. S. W. G. of the Psychiatric Genomics Consortium et al. Biological insights from 108 schizophrenia-associated genetic loci. *Nature*, 511(7510):421–427, 2014.
48. T. Park and G. Casella. The Bayesian lasso. *J. Am. Stat. Assoc.*, 103:681–688, 2008.
49. L. Pasanen, L. Holmström, and M. J. Sillanpää. Bayesian lasso, scale space and decision making in association genetics. *PLoS ONE*, 10(4):e0120017, 2015.
50. P. C. Phillips. Epistasis—the essential role of gene interactions in the structure and evolution of genetic systems. *Nature Reviews Genetics*, 9(11):855–867, 2008.
51. A. Rahimi, B. Recht, et al. Random features for large-scale kernel machines. In *NIPS*, volume 3, page 5, 2007.
52. J. C. Randall, T. W. Winkler, Z. Kutalik, S. I. Berndt, A. U. Jackson, K. L. Monda, T. O. Kilpeläinen, T. Esko, R. Mägi, S. Li, T. Workalemahu, M. F. Feitosa, D. C. Croteau-Chonka, F. R. Day, T. Fall, T. Ferreira, S. Gustafsson, A. E. Locke, I. Mathieson, A. Scherag, S. Vedantam, A. R. Wood, L. Liang, V. Steinthorsdottir, G. Thorleifsson, E. T. Dermitzakis, A. S. Dimas, F. Karpe, J. L. Min, G. Nicholson, D. J. Clegg, T. Person, J. P. Krohn, S. Bauer, C. Buechler, K. Eisinger, D. Consortium, A. Bonnefond, P. Froguel, M. Investigators, J.-J. Hottenga, I. Prokopenko, L. L. Waite, T. B. Harris, A. V. Smith, A. R. Shuldiner, W. L. McArdle, M. J. Caulfield, P. B. Munroe, H. Grönberg, Y.-D. I. Chen, G. Li, J. S. Beckmann, T. Johnson, U. Thorsteinsdottir, M. Teder-Laving, K.-T. Khaw, N. J. Wareham, J. H. Zhao, N. Amin, B. A. Oostra, A. T. Kraja, M. A. Province, L. A. Cupples, N. L. Heard-Costa, J. Kaprio, S. Ripatti, I. Surakka, F. S. Collins, J. Saramies, J. Tuomilehto, A. Jula, V. Salomaa, J. Erdmann, C. Hengstenberg, C. Loley, H. Schunkert, C. Lamina, H. E. Wichmann, E. Albrecht, C. Gieger, A. A. Hicks, Å. Johansson, P. P. Pramstaller, S. Kathiresan, E. K. Speliotes, B. Penninx, A.-L. Hartikainen, M.-R. Jarvelin, U. Gyllensten, D. I. Boomsma, H. Campbell, J. F. Wilson, S. J. Chanock, M. Farrall, A. Goel, C. Medina-Gomez, F. Rivadeneira, K. Estrada, A. Uitterlinden, A. Hofman, M. C. Zillikens, M. den Heijer, L. A. Kiemeney, A. Maschio, P. Hall, J. Tyrer, A. Teumer, H. Völzke, P. Kovacs, A. Tönjes, M. Mangino, T. D. Spector, C. Hayward, I. Rudan, A. S. Hall, N. J. Samani, A. P. Attwood, J. G. Sambrook, J. Hung, L. J. Palmer, M.-L. Lokki, J. Sinisalo, G. Boucher, H. Huikuri, M. Lorentzon, C. Ohlsson, N. Eklund,

J. G. Eriksson, C. Barlassina, C. Rivolta, I. M. Nolte, H. Snieder, M. M. Van der Klauw, J. V. Van Vliet-Ostaptchouk, P. V. Gejman, J. Shi, K. B. Jacobs, Z. Wang, S. J. L. Bakker, I. Mateo Leach, G. Navis, P. van der Harst, N. G. Martin, S. E. Medland, G. W. Montgomery, J. Yang, D. I. Chasman, P. M. Ridker, L. M. Rose, T. Lehtimäki, O. Raitakari, D. Absher, C. Iribarren, H. Basart, K. G. Hovingh, E. Hyppönen, C. Power, D. Anderson, J. P. Beilby, J. Hui, J. Jolley, H. Sager, S. R. Bornstein, P. E. H. Schwarz, K. Kristiansson, M. Perola, J. Lindström, A. J. Swift, M. Uusitupa, M. Atalay, T. A. Lakka, R. Rauramaa, J. L. Bolton, G. Fowkes, R. M. Fraser, J. F. Price, K. Fischer, K. KrjutÅ!'kov, A. Metspalu, E. Mihailov, C. Langenberg, J. Luan, K. K. Ong, P. S. Chines, S. M. Keinanen-Kiukaanniemi, T. E. Saaristo, S. Edkins, P. W. Franks, G. Hallmans, D. Shungin, A. D. Morris, C. N. A. Palmer, R. Erbel, S. Moebus, M. M. Nöthen, S. Pechlivanis, K. Hveem, N. Narisu, A. Hamsten, S. E. Humphries, R. J. Strawbridge, E. Tremoli, H. Grallert, B. Thorand, T. Illig, W. Koenig, M. Müller-Nurasyid, A. Peters, B. O. Boehm, M. E. Kleber, W. März, B. R. Winkelmann, J. Kuusisto, M. Laakso, D. Arveiler, G. Cesana, K. Kuulasmaa, J. Virtamo, J. W. G. Yarnell, D. Kuh, A. Wong, L. Lind, U. de Faire, B. Gigante, P. K. E. Magnusson, N. L. Pedersen, G. Dedoussis, M. Dimitriou, G. Kolovou, S. Kanoni, K. Stirrups, L. L. Bonnycastle, I. Njølstad, T. Wilsgaard, A. Ganna, E. Rehnberg, A. Hingorani, M. Kivimaki, M. Kumari, T. L. Assimes, I. Barroso, M. Boehnke, I. B. Borecki, P. Deloukas, C. S. Fox, T. Frayling, L. C. Groop, T. Haritunians, D. Hunter, E. Ingelsson, R. Kaplan, K. L. Mohlke, J. R. O'Connell, D. Schlessinger, D. P. Strachan, K. Stefansson, C. M. van Duijn, G. R. Abecasis, M. I. McCarthy, J. N. Hirschhorn, L. Qi, R. J. F. Loos, C. M. Lindgren, K. E. North, and I. M. Heid. Sex-stratified genome-wide association studies including 270,000 individuals show sexual dimorphism in genetic loci for anthropometric traits. *PLOS Genetics*, 9(6):e1003500–, 06 2013.

53. W. Rudin. *Fourier analysis on groups*. John Wiley & Sons, 2011.
54. B. Scholkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.
55. M. Scutari, I. Mackay, and D. Balding. Improving the efficiency of genomic selection. *Statistical applications in genetics and molecular biology*, 12(4):517–527, 2013.
56. Q. Shi, J. Petterson, G. Dror, J. Langford, A. Smola, and S. Vishwanathan. Hash kernels for structured data. *Journal of Machine Learning Research*, 10(Nov):2615–2637, 2009.
57. P. Sollich. Bayesian methods for support vector machines: Evidence and predictive class probabilities. *Machine learning*, 46(1-3):21–52, 2002.
58. D. Speed and D. J. Balding. Multiblup: improved snp-based prediction for complex traits. *Genome research*, 24(9):1550–1557, 2014.
59. W. Valdar, L. C. Solberg, D. Gauguier, S. Burnett, P. Klenerman, W. O. Cookson, M. S. Taylor, J. N. P. Rawlins, R. Mott, and J. Flint. Genome-wide genetic association of complex traits in heterogeneous stock mice. *Nature genetics*, 38(8):879–887, 2006.
60. W. Valdar, L. C. Solberg, D. Gauguier, W. O. Cookson, J. N. P. Rawlins, R. Mott, and J. Flint. Genetic and environmental effects on complex traits in mice. *Genetics*, 174(2):959–984, 2006.
61. P. M. Visscher, M. A. Brown, M. I. McCarthy, and J. Yang. Five years of gwas discovery. *The American Journal of Human Genetics*, 90(1):7–24, 2012.

62. G. Wahba et al. Support vector machines, reproducing kernel hilbert spaces and the randomized gacv. *Advances in Kernel Methods-Support Vector Learning*, 6:69–87, 1999.
63. D. Wang, I. S. El-Basyoni, P. S. Baenziger, J. Crossa, K. M. Eskridge, and I. Dweikat. Prediction of genetic values of quantitative traits with epistatic effects in plant breeding populations. *Heredity*, 109(5):313–319, 2012.
64. W.-H. Wei, G. Hemani, and C. S. Haley. Detecting epistasis in human complex traits. *Nature Reviews Genetics*, 15(11):722–733, 2014.
65. K. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg. Feature hashing for large scale multitask learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1113–1120. ACM, 2009.
66. J. Yang, B. Benyamin, B. P. McEvoy, S. Gordon, A. K. Henders, D. R. Nyholt, P. A. Madden, A. C. Heath, N. G. Martin, G. W. Montgomery, et al. Common snps explain a large proportion of the heritability for human height. *Nature genetics*, 42(7):565–569, 2010.
67. J. Yang, S. H. Lee, M. E. Goddard, and P. M. Visscher. Gcta: a tool for genome-wide complex trait analysis. *The American Journal of Human Genetics*, 88(1):76–82, 2011.
68. A. Zeileis, K. Hornik, A. Smola, and A. Karatzoglou. kernlab-an s4 package for kernel methods in r. *Journal of statistical software*, 11(9):1–20, 2004.
69. Y. Zhao, M. F. Mette, and J. C. Reif. Genomic selection in hybrid breeding. *Plant Breeding*, 134(1):1–10, 2015.
70. X. Zhou, P. Carbonetto, and M. Stephens. Polygenic modeling with Bayesian sparse linear mixed models. *PLoS Genet*, 9(2):e1003264, 02 2013.
71. O. Zuk, E. Hechter, S. R. Sunyaev, and E. S. Lander. The mystery of missing heritability: Genetic interactions create phantom heritability. *Proceedings of the National Academy of Sciences*, 109(4):1193–1198, 2012.