Doctoral theses at NTNU, 2022:26

Pål Vegard Johnsen

Explainability and validity of statistical methods for genome-wide association studies:

Extending Shapley-based explanation methods and adapting saddlepoint approximations

Doctoral thesis

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of
Philosophiae Doctor
Faculty of Information Technology and Electrical
Engineering
Department of Mathematical Sciences

NTNU
Norwegian University of
Science and Technology

Pål Vegard Johnsen

Explainability and validity of statistical methods for genome-wide association studies:

Extending Shapley-based explanation methods and adapting saddlepoint approximations

Thesis for the Degree of Philosophiae Doctor

Trondheim, January 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

© Pål Vegard Johnsen

ISBN 978-82-326-5639-4 (printed ver.)
ISBN 978-82-326-6718-5 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

Doctoral theses at NTNU, 2022:26

Printed by NTNU Grafisk senter

Acknowledgements

I am so gratified and grateful that I got my chance to get a PhD degree. This work would not have been possible without so many kind-hearted people. I would like to thank

- My supervisor Mette Langaas for your endless capacity and will of helping me whenever I needed it. Your support has been beyond perfect. Thank you so much for all the great discussions we have had, and everything I have learned from you.
- My co-supervisor Signe Riemer-Sørensen for joyful and efficient discussions. Thank you very much for all your support, your bright brain and your never ending optimism. It has been a very pleasant journey to work with you.
- Øyvind Bakke and Thea Bjørnland for the great group meetings we have had with so much fruitful and joyful discussions about the thing we love: Statistics. You have been of great support, and I really respect your great competence.
- Andrew Thomas DeWan and Megan Cahill for the wonderful period I had at Yale School of Public Health. I enjoyed the time in New Haven, and I appreciate the great collaboration we had.
- Tormod Rogne for being an excellent colleague of brilliant competence. Thank you so much for all the discussions and your help.
- My girlfriend, Veerle, for always being there for me. I will never forget the beautiful period we had together during the frightening COVID-19 lockdown. Thanks to you I kept my focus and motivation.
- My mom and my sister, Annette, for always supporting me no matter what at any time. Thank you so much for your big hearts, kindness and enormous empathy.
- Last, but not least, I would like to thank my dad. Sadly you did not get the chance to experience the end of my PhD term. It would have been so great if you would be able to see your son get the PhD that you could not afford yourself, but really wanted, in your early academic life. I would never been able to do a PhD without all your endless support for all those years. I know you would have been very proud of me no matter what, and I choose to believe that you are somewhere out there wishing me good luck.

List of research papers

Paper 1: Saddlepoint approximations in binary genome-wide association studies
Pål V. Johnsen, Øyvind Bakke, Thea Bjørnland, Andrew Thomas DeWan, and Mette Langaas
2021, arXiv: https://arxiv.org/abs/2110.04025

Paper 2: A new method for exploring gene–gene and gene–environment interactions in GWAS with tree ensemble methods and SHAP values
Pål V. Johnsen, Signe Riemer-Sørensen, Andrew Thomas DeWan, Megan E. Cahill and Mette Langaas
BMC Bioinformatics, 2021 22
https://doi.org/10.1186/s12859-021-04041-7

Paper 3: Inferring feature importance with uncertainties in high-dimensional data
Pål V. Johnsen, Inga Strümke, Signe Riemer-Sørensen, Andrew Thomas DeWan and Mette Langaas
2021, arXiv: https://arxiv.org/abs/2109.00855

Contents

Part I

Thesis Summary

1 Introduction

This thesis is submitted as a collection of three papers, from hereon denoted Paper 1, Paper 2 and Paper 3. All three papers are connected to so-called *genome-wide association studies* (GWAS), and statistical challenges that arise in this particular setting. In short, the aim of a GWAS is to infer whether there is an *association* between a *genetic marker* and a *trait* for a particular organism. We will in this thesis work with human data. The genetic marker in this particular setting is a so-called *single nucleotide polymorphism* (SNP) to be in Section 2. A trait is a particular feature of a given species. This can be weight, height, blood pressure or presence or absence of a disease such as obesity or sepsis. A *phenotype* is a particular observation, or outcome, of a given trait. This is for instance a height of 180 cm, or the observation of a BMI greater than 30, in which case the particular individual will be defined as obese. In this thesis we will restrict ourselves to traits with two possible outcomes such as presence or absence of a disease, and we will refer to this as a trait with binary outcomes.

Assuming a set of unrelated individuals, the statistical approach most used in GWAS is by construction of generalized linear regression models for each SNP, and then inference of statistically significant associations via statistical tests such as the likelihood ratio test, the Wald test or the score test (Casella & Berger, 2001). Calculation of the corresponding *p*-values can often be achieved accurately by normal approximation of the test statistic. However, for traits with only two possible outcomes, there are circumstances in which normal approximation is not sufficiently accurate, consequently resulting in *invalid* *p*-values. It turns out that another type of approximation is even more accurate than the normal approximation, namely the so-called *saddlepoint approximation* (Dey et al., 2017; Ma et al., 2013). In fact, there are several possible approaches to do saddlepoint approximation of a probability distribution. In Paper 1, the use of saddlepoint approximations for GWAS is investigated thoroughly to see in which cases the corresponding *p*-value is considered valid.

The aim of a GWAS to infer association between a single SNP and a trait can be further generalized. There is reason to believe that the association of a SNP and a trait may depend on some other SNP, not simply because of potential correlation between the two SNPs, but because the effect of the two SNPs on the trait is intertwined. In other words, the total effect of the two SNPs on the trait is not simply the additive effect of the two SNPs. In this case, one says there is an *interaction* between the two SNPs with respect to the associative effect on the trait. Likewise, one can say there is an interaction between a SNP and a so-called *environmental covariate.* Examples of environmental covariates are age, sex and smoking status or any other covariate not connected to one simple SNP. Inferring interactions has turned out to be particularly challenging when applying classical generalized linear models with *interaction* terms. One reason for this is due to the strict, but necessary, rule for declaring the interaction as statistically significant. This is in general called the *multiple comparison problem* to be discussed in Section 4. Another possible reason may be that the constructed regression models are not sufficiently complex to find such interactions. At the same time, there has been several breakthroughs within machine learning from the end of 20th century, and until now. Machine learning models such as deep neural networks, and tree-ensemble models have been shown to be strong predictive models, and advances in computer science made sure that the models could be efficiently constructed within feasible time frames and storage capacities, even when based on high-dimensional data. At the same time, these complex models are often referred to as *black box models*, as they are far from as interpretable as linear regression models. The question then arises whether such models still have the capacity to find interactions in a GWAS setting, and if so how to interpret these black box models. This is the aim of Paper 2, where it is investigated how one such black box model, a tree-ensemble model, can be used in order to interpret and find possible interaction candidates in a GWAS setting specifically for a trait with binary outcomes.

Even though black box models may find more complex relationships than generalized regression models, the complexity of these models makes it more challenging to *infer* whether for instance an interaction candidate actually is a true interaction. While one can often achieve, at least asymptotically, reasonable approximations of the distribution of the test statistics applied in generalized linear models, the same can not be said when applying Shapley based procedures to interpret black box models. The question of how to infer whether covariate effects are significant (non-zero) or not in black box models has been an increasingly popular topic within the machine learning community. This is particularly challenging in high-dimensional data. In Paper 3, a Shapley based feature importance measure is proposed, together with a bootstrap procedure to estimate the uncertainty in the corresponding feature importance estimator.

Before presenting the three papers that constitute the research contribution of the thesis, a brief introduction of central topics is given.

2 The human genome, SNPs, genotyping and UK biobank

Recall that for human beings the genome consists of 23 chromosome pairs, where each chromosome includes a DNA molecule with its characteristic double helix structure. Each of the two strands in the double helix consists of a sequence of repeated molecular units called *nucleotides*. A nucleotide includes a *nucleobase*, or simply a base, and the two strands are connected as a result of hydrogen bonds between nucleobases from each strand. There are only four possible bases: Adenine (A), cytocine (C), guanine (G) and thymine (T). Moreover, two bases are bounded in only two ways: Adenine with thymine (A-T), and cytocine with guanine (C-G). Apart from the two chromosomes deciding the sex, each of the 22 other pairs, called *autosomal* chromosomes, consists of two identical chromosomes of same size. Hence, a specific *position* in the genome refers to a base-pair position in a particular chromosome. In around 0.01 % of the in total 3.2 billion base-pair positions in the genome, there is variation in the type of base-pair defined as a *single nucleotide polymorphism* (SNP). Variation in a base-pair position is defined as a SNP variant if the least frequent base-pair arises in more than 1 % of the population. A specific base-pair in such a position is often denoted an *allele*. Most SNPs are *biallelic*, meaning there are only two possible alleles. The *minor allele frequency* (MAF) is the proportion of the least frequent allele, in a population. For instance with N individuals and m observations of the least frequent allele, the MAF is estimated to be $m/2N$ since there are two copies of each chromosome in each individual. Variants with MAF less than 1 % are called *rare variants*. In this thesis, the focus will be on biallelic variants.

In order to investigate the SNPs in the human genome and to do a GWAS, the genetic data is produced by so-called *genotyping arrays*. Advances in the technology has enabled low production cost for high-accuracy genetic data. As a result, for each individual, the exact base-pair is investigated in several selected base-pair positions spread along the whole genome. The data produced for each base-pair position, and each individual is given as a *minor allele count*, sometimes also called the genotype value. This is simply the count of the minor allele in the given base-pair position, by looking at both of the paired chromosomes. As there in most cases are only two possible alleles, the count goes from zero to two. Typically the resulting SNP data includes hundreds of thousands of SNPs spread along the whole genome.

UK biobank is a large-scale long-term biomedical database consisting of around 500 000 British participants, and available for all bona fide researchers via application (Bycroft et al., 2018). As the participants are followed up also after initial assessment, it is called a *prospective cohort study*. The individuals were aged 40-69 years when they joined UK Biobank in the period 2006-2010. For each of the 500 000 participants, their genome has been genotyped. Along with genetic data, the biobank also consists of a vast amount of clinical data collected during initial assessment such as sex, smoking status and amount of physical activity to mention a few. In addition, records of hospitalizations before and after initial assessment is available for each individual given as ICD-codes (ICD-9 and ICD-10). The ICD, abbreviated for International Statistical Classification of Diseases and Related Health Problems, is an international register with a specific code for each disease as well as symptoms. Consequently, this makes the biobank one of the most popular data sources to conduct a GWAS. In all three papers of this thesis, UK Biobank is applied when conducting the research.

3 Score vector

The *score vector*, $\boldsymbol{U}$, with respect to the random vector $\mathbf{Y}$ of size n and parameters $\boldsymbol{\theta}$ is by definition the gradient of the loglikelihood function with respect to $\boldsymbol{\theta}$:

$$\boldsymbol{U} = \nabla_{\boldsymbol{\theta}} \ln L(\boldsymbol{\theta}|\boldsymbol{Y}) = \frac{\partial}{\partial \boldsymbol{\theta}} \ln L(\boldsymbol{\theta}|\boldsymbol{Y}) = \frac{\partial}{\partial \boldsymbol{\theta}} \ell(\boldsymbol{\theta}|\mathbf{Y}),$$

where the likelihood function $L(\boldsymbol{\theta} \mid \boldsymbol{y}) = f_{\boldsymbol{Y}}(\boldsymbol{y} \mid \boldsymbol{\theta})$, the joint probability distribution of the random vector $\boldsymbol{Y}$. Observe that we may write according to the rule of derivative of natural logarithms:

$$E(\boldsymbol{U}) = E\left(\frac{\partial}{\partial \boldsymbol{\theta}}\ell(\boldsymbol{\theta}|\mathbf{Y})\right) = E\Big(\frac{\partial}{\partial \boldsymbol{\theta}}\ln f(\mathbf{Y}|\boldsymbol{\theta})\Big) = E\left(\frac{\frac{\partial}{\partial \boldsymbol{\theta}}f(\mathbf{Y}|\boldsymbol{\theta})}{f(\mathbf{Y}|\boldsymbol{\theta})}\right). \tag{1}$$

If $\boldsymbol{Y}$ has a discrete distribution it can easily be seen that $E_{\boldsymbol{\theta}}(\boldsymbol{U}) = \mathbf{0}$:

$$E(\boldsymbol{U}) = E\left(\frac{\frac{\partial}{\partial \boldsymbol{\theta}}f(\mathbf{Y}|\boldsymbol{\theta})}{f(\mathbf{Y}|\boldsymbol{\theta})}\right) = \sum_{\mathbf{y}} \frac{\partial}{\partial \boldsymbol{\theta}} f(\mathbf{y}|\boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}} \sum_{\mathbf{y}} f(\mathbf{y}|\boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}}\mathbf{1} = \mathbf{0},$$

by the linearity property of differentiation. If $\boldsymbol{Y}$ is continuous:

$$E(\boldsymbol{U}) = E\left(\frac{\frac{\partial}{\partial \boldsymbol{\theta}}f(\mathbf{Y}|\boldsymbol{\theta})}{f(\mathbf{Y}|\boldsymbol{\theta})}\right) = \int_{\mathbf{y}} \frac{\partial}{\partial \boldsymbol{\theta}} f(\mathbf{y}|\boldsymbol{\theta}) d\mathbf{y}.$$

In the cases where the following applies:

$$\int_{\mathbf{y}} \frac{\partial}{\partial \boldsymbol{\theta}} f(\mathbf{y}|\boldsymbol{\theta}) d\mathbf{y} = \frac{\partial}{\partial \boldsymbol{\theta}} \int_{\mathbf{y}} f(\mathbf{y}|\boldsymbol{\theta}) d\mathbf{y}, \tag{2}$$

$E(\boldsymbol{U}) = \mathbf{0}$ in this case as well. This property holds for all parametric probability distributions within the *exponential family*. By having such a probability distribution, the covariance of the score vector:

$$\begin{aligned}
&\text{Cov}(\boldsymbol{U}) = E_{\boldsymbol{\theta}}\left(\boldsymbol{U}\boldsymbol{U}^T\right) - E_{\boldsymbol{\theta}}\left(\boldsymbol{U}\right) E_{\boldsymbol{\theta}}\left(\boldsymbol{U}\right)^T = E_{\boldsymbol{\theta}}\left(\boldsymbol{U}\boldsymbol{U}^T\right) \\
&= E\begin{pmatrix}
(\frac{\partial}{\partial \theta_1}\ell(\boldsymbol{\theta}|\mathbf{Y}))^2 & \cdots & \cdots & (\frac{\partial}{\partial \theta_1}\ell(\boldsymbol{\theta}|\mathbf{Y}))(\frac{\partial}{\partial \theta_p}\ell(\boldsymbol{\theta}|\mathbf{Y})) \\
(\frac{\partial}{\partial \theta_2}\ell(\boldsymbol{\theta}|\mathbf{Y}))(\frac{\partial}{\partial \theta_1}\ell(\boldsymbol{\theta}|\mathbf{Y})) & (\frac{\partial}{\partial \theta_2}\ell(\boldsymbol{\theta}|\mathbf{Y}))^2 & \cdots & (\frac{\partial}{\partial \theta_2}\ell(\boldsymbol{\theta}|\mathbf{Y}))(\frac{\partial}{\partial \theta_p}\ell(\boldsymbol{\theta}|\mathbf{Y})) \\
\vdots & \vdots & \ddots & \vdots \\
(\frac{\partial}{\partial \theta_p}\ell(\boldsymbol{\theta}|\mathbf{Y}))(\frac{\partial}{\partial \theta_1}\ell(\boldsymbol{\theta}|\mathbf{Y})) & \cdots & \cdots & (\frac{\partial}{\partial \theta_p}\ell(\boldsymbol{\theta}|\mathbf{Y}))^2
\end{pmatrix}.
\end{aligned} \tag{3}$$

For some probability distributions, such as those within the exponential family, one can show that:

$$E\left(\left(\frac{\partial}{\partial \theta_j}\ell\left(\boldsymbol{\theta}|\mathbf{Y}\right)\right)\left(\frac{\partial}{\partial \theta_k}\ell\left(\boldsymbol{\theta}|\mathbf{Y}\right)\right)\right) = -E\left(\frac{\partial}{\partial \theta_j}\frac{\partial}{\partial \theta_k}\ell\left(\boldsymbol{\theta}|\mathbf{Y}\right)\right).$$

With this property we refer the covariance of the score vector as the expected *Fisher information*, $F(\boldsymbol{\theta})$, given by:

$$F(\boldsymbol{\theta}) = -E\begin{pmatrix}
\frac{\partial^2}{\partial \theta_1^2}\ell(\boldsymbol{\theta}|\mathbf{Y}) & \cdots & \cdots & \frac{\partial^2}{\partial \theta_1 \partial \theta_p}\ell(\boldsymbol{\theta}|\mathbf{Y}) \\
\frac{\partial^2}{\partial \theta_2 \partial \theta_1}\ell(\boldsymbol{\theta}|\mathbf{Y}) & \frac{\partial^2}{\partial \theta_2^2}\ell(\boldsymbol{\theta}|\mathbf{Y}) & \cdots & \frac{\partial^2}{\partial \theta_2 \partial \theta_p}\ell(\boldsymbol{\theta}|\mathbf{Y}) \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial^2}{\partial \theta_p \partial \theta_1}\ell(\boldsymbol{\theta}|\mathbf{Y}) & \cdots & \cdots & \frac{\partial^2}{\partial \theta_p^2}\ell(\boldsymbol{\theta}|\mathbf{Y}).
\end{pmatrix} \tag{4}$$

From hereon, we consider the elements of the random vector $\boldsymbol{Y}$, denoted $Y_1, \ldots, Y_n$, to be identically distributed and independent random variables. Then $L(\boldsymbol{\theta} \mid \boldsymbol{y}) = \prod_{i=1}^N f_Y(y_i \mid \boldsymbol{\theta})$, where $f_Y(y; \boldsymbol{\theta})$ is the probability distribution of Y with parameter vector $\boldsymbol{\theta}$, and therefore $\ell(\boldsymbol{\theta}|\mathbf{y}) = \sum_{i=1}^n \ln f_Y(y_i; \boldsymbol{\theta})$, in which each element of the score vector, U_j, for $j = 1, \ldots, p$ is given by $U_j = \sum_{i=1}^n \frac{\partial}{\partial \theta_j} \ln f_Y(Y_i; \boldsymbol{\theta})$. By, the central limit theorem as $n \to \infty$, the score vector $\boldsymbol{U}$ will asymptotically have a multivariate normal distribution:

$$\boldsymbol{U} \xrightarrow{d} N(\mathbf{0}, F(\boldsymbol{\theta})).$$

We define the *observed* Fisher Information, $\mathcal{I}(\boldsymbol{\theta})$:

$$\mathcal{I}(\boldsymbol{\theta}) = -\begin{pmatrix} \frac{\partial^2}{\partial \theta_1^2}\ell(\boldsymbol{\theta}|\mathbf{Y}) & \cdots & \cdots & \frac{\partial^2}{\partial \theta_1 \partial \theta_p}\ell(\boldsymbol{\theta}|\mathbf{Y}) \\ \frac{\partial^2}{\partial \theta_2 \partial \theta_1}\ell(\boldsymbol{\theta}|\mathbf{Y}) & \frac{\partial^2}{\partial \theta_2^2}\ell(\boldsymbol{\theta}|\mathbf{Y}) & \cdots & \frac{\partial^2}{\partial \theta_2 \partial \theta_p}\ell(\boldsymbol{\theta}|\mathbf{Y}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2}{\partial \theta_p \partial \theta_1}\ell(\boldsymbol{\theta}|\mathbf{Y}) & \cdots & \cdots & \frac{\partial^2}{\partial \theta_p^2}\ell(\boldsymbol{\theta}|\mathbf{Y}) \end{pmatrix} \tag{5}$$

By the (strong) *law of large numbers*, the observed Fisher information, $\mathcal{I}(\boldsymbol{\theta})$, converges almost surely (and therefore in probability) to the true Fisher information (4). Hence, one can show that:

$$\boldsymbol{U} \xrightarrow{d} N(\mathbf{0}, \mathcal{I}(\boldsymbol{\theta})).$$

Consider the partition of the parameter $\boldsymbol{\theta} = (\boldsymbol{\beta}^T, \boldsymbol{\gamma}^T)^T$ consisting of the parameter vector $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, and let $\boldsymbol{U}_\beta$ and $\boldsymbol{U}_{\boldsymbol{\gamma}}$ denote the corresponding partition of the score vector $\boldsymbol{U}$. In this case we may write the observed Fischer information as:

$$\mathcal{I}(\boldsymbol{\theta}) = -\begin{pmatrix} \frac{\partial^2}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T}\ell(\boldsymbol{\theta}|\mathbf{Y}) & \frac{\partial^2}{\partial \boldsymbol{\beta} \partial \boldsymbol{\gamma}^T}\ell(\boldsymbol{\theta}|\mathbf{Y}) \\ \frac{\partial^2}{\partial \boldsymbol{\gamma} \partial \boldsymbol{\beta}^T}\ell(\boldsymbol{\theta}|\mathbf{Y}) & \frac{\partial^2}{\partial \boldsymbol{\gamma} \partial \boldsymbol{\gamma}^T}\ell(\boldsymbol{\theta}|\mathbf{Y}) \end{pmatrix} = \begin{pmatrix} \mathcal{I}_{\boldsymbol{\beta\beta}} & \mathcal{I}_{\boldsymbol{\beta\gamma}} \\ \mathcal{I}_{\boldsymbol{\gamma\beta}} & \mathcal{I}_{\boldsymbol{\gamma\gamma}}, \end{pmatrix} \tag{6}$$

where $\mathcal{I}_{\boldsymbol{\beta\beta}}$, $\mathcal{I}_{\boldsymbol{\gamma\gamma}}$ and $\mathcal{I}_{\boldsymbol{\beta\gamma}} = \mathcal{I}_{\boldsymbol{\gamma\beta}}^T$ are submatrices of the observed Fisher information in (5) involving the particular parameters of interest. Using this notation, and by the property of multivariate normal distributions, one can show that the *conditional distribution* $f(\boldsymbol{u}_\gamma \mid \boldsymbol{U}_\beta = \boldsymbol{u}_\beta)$ asymptotically will have a multivariate normal distribution:

$$\boldsymbol{U}_{\boldsymbol{\gamma}} \mid (\boldsymbol{U}_{\boldsymbol{\beta}} = \boldsymbol{u}_{\boldsymbol{\beta}}) \xrightarrow{d} N\left(\mathcal{I}_{\boldsymbol{\beta\gamma}}\mathcal{I}_{\boldsymbol{\beta\beta}}^{-1}\boldsymbol{u}_\beta, \mathcal{I}_{\boldsymbol{\gamma\gamma}} - \mathcal{I}_{\boldsymbol{\gamma\beta}}\mathcal{I}_{\boldsymbol{\beta\beta}}^{-1}\mathcal{I}_{\boldsymbol{\beta\gamma}}\right).$$

Consider the hypothesis test

$$H_0 \colon \boldsymbol{\gamma} = 0 \quad \text{against} \quad H_1 \colon \gamma \neq 0.$$

Assuming the parameters $\boldsymbol{\beta}$ are unknown, we denote these as nuisance parameters. We denote $\hat{\boldsymbol{\beta}}$ as the maximum likelihood estimates of $\boldsymbol{\beta}$ when the null hypothesis is true, with the estimator by definition given by the equation $\boldsymbol{U}_{\boldsymbol{\beta}} = \mathbf{0}$. Let $\hat{\boldsymbol{\theta}} = (\hat{\boldsymbol{\beta}}^T, \mathbf{0}^T)^T$ denote the corresponding parameter estimate under the null hypothesis. Assuming the null hypothesis to be true, and due to the *consistency* of maximum likelihood estimators, one can show that (Lindsey, 1996):

$$\boldsymbol{U}_{\boldsymbol{\gamma}}(\boldsymbol{\theta}) \mid (\boldsymbol{U}_{\boldsymbol{\beta}} = \mathbf{0}) \xrightarrow{d} N\left(\mathbf{0}, \mathcal{I}(\hat{\boldsymbol{\theta}})_{\boldsymbol{\gamma\gamma}} - \mathcal{I}(\hat{\boldsymbol{\theta}})_{\boldsymbol{\gamma\beta}}\mathcal{I}(\hat{\boldsymbol{\theta}})_{\boldsymbol{\beta\beta}}^{-1}\mathcal{I}(\hat{\boldsymbol{\theta}})_{\boldsymbol{\beta\gamma}}\right). \tag{7}$$

4 GWAS

The usual set-up when performing a GWAS, assuming unrelated individuals, and based on SNP array data is to first construct a generalized regression model *for each SNP*, including the allele count of the SNP as well as additional covariates, sometimes also called *environmental* covariates, such as intercept, age, sex or smoking status. Denote $\boldsymbol{g}$ the *genotype* vector including the allele count, g_i, for each individual i for a total of N individuals. Moreover, let $\boldsymbol{x}_i$ denote the vector of $d-1$ covariates, including intercept, for individual i. Furthermore, let $\boldsymbol{Y}$ denote the random vector of *responses* with probability distribution according to the generalized regression model:

$$g(\mu_i) = \boldsymbol{x}_i^T\boldsymbol{\beta} + \gamma g_i, \tag{8}$$

where the function $g()$ is such that $g^{-1}(\boldsymbol{x}_i^T\boldsymbol{\beta} + \gamma g_i) = \mu_i = E[Y_i \mid \mathbf{x}_i, g_i]$. Two examples are the identity function $g(\mu_i) = \mu_i$ which gives the classical multiple regression model, and $g(\mu_i) = \text{logit}(\mu_i)$ with

$$\text{logit}(\mu_i) = \ln\left(\frac{\mu_i}{1-\mu_i}\right), \tag{9}$$

which gives the logistic regression model. Furthermore, $\boldsymbol{\beta}$ is the nuisance parameter vector of size $d-1$, while γ is the parameter of interest in the sense that we want to perform the following hypothesis test:

$$H_0 \colon \gamma = 0 \quad \text{against} \quad H_1 \colon \gamma \neq 0. \tag{10}$$

In other words, we want to test whether there is an association between a particular SNP and the trait of interest.

In this thesis, the focus will be on logistic regression models, meaning that the response, Y_i, is either zero or one, hence having a *Bernoulli* distribution:

$$f(y_i) = \mu_i^{y_i}(1-\mu_i)^{1-y_i}.$$

with $\mu_i = E[Y_i \mid \mathbf{x}_i, g_i] = P(Y_i = 1 \mid \mathbf{x}_i, g_i)$, modelled as in (8) with identity function as in (9). We will from hereon denote this as a *binary* GWAS. In most cases, the purpose of a binary GWAS is to investigate the presence or absence of a certain disease or symptom, and usually $Y_i = 1$ denotes the presence of a disease for individual i. There are typically two strategies when collecting the data in a binary GWAS. In a *case-control study*, we first collect individuals that we know have a disease, called cases, as well as individuals that are thought to never have had the disease, denoted controls. The individuals are chosen in order to avoid biases such as making sure that the groups come from similar populations. Afterwards, we collect environmental covariates as well as collect genetic samples in order to create SNP data. In a *prospective cohort study*, individuals from a given population are selected at a time point and followed up in a given period of time. During this time interval, the health conditions of the individuals are recorded, perhaps limited to certain diseases or symptoms of interest. This allows us to investigate the presence of a disease for each individual during the time period. The environmental covariates as well as genetic samples are collected already during recruitment of individuals.

Both case-control studies and prospective cohort studies are equally valid procedures to conduct a binary GWAS. In this thesis, we use the UK Biobank, a prospective cohort study. In that case, the interpretation of what we try to model, *for each SNP*, is: What is the probability that a given individual will get a certain disease within a given period of time given information about the allele count of the SNP and the covariates.

4.1 Application of the score test statistic in binary GWAS

For each SNP, the hypothesis test in (10) can be evaluated by using well known test statistics such as the likelihood ratio test statistic, the Wald test statistic or the score test statistic, all based on maximum likelihood (ML) theory. The likelihood ratio test (LRT) requires the corresponding ML estimates under both the null hypothesis, H_0, as well as the alternative hypothesis, H_1. For the Wald test, we require the ML estimates only under the alternative hypothesis. For the score test, we only require the estimates under the null hypothesis. This property of the score test statistic is particularly convenient when doing a large number of hypothesis tests, such as in GWAS. Practically it means that the ML estimates under the null hypothesis need only be computed once for all SNPs since the null hypothesis is exactly the same for each SNP.

The score vector, $\boldsymbol{U}$, for the Bernoulli distribution modelled as in (8) with parameters $\boldsymbol{\theta} = (\boldsymbol{\beta}^T, \gamma)^T$ is given by:

$$\begin{aligned}
\boldsymbol{U} &= \begin{pmatrix} \frac{\partial}{\partial \boldsymbol{\beta}} \ell(\boldsymbol{\theta} \mid \boldsymbol{Y}) \\ \frac{\partial}{\partial \gamma} \ell(\boldsymbol{\theta} \mid \boldsymbol{Y}) \end{pmatrix} = \begin{pmatrix} \frac{\partial}{\partial \boldsymbol{\beta}} \sum_{i=1}^n Y_i \ln(\mu_i) + (1-Y_i)\ln(1-\mu_i) \\ \frac{\partial}{\partial \gamma} \sum_{i=1}^n Y_i \ln(\mu_i) + (1-Y_i)\ln(1-\mu_i) \end{pmatrix} \\
&= \begin{pmatrix} \frac{\partial}{\partial \boldsymbol{\beta}} \sum_{i=1}^n Y_i(\boldsymbol{x}_i^T \boldsymbol{\beta} + \gamma g_i) - \ln(1+\exp(\boldsymbol{x}_i^T \boldsymbol{\beta} + \gamma g_i)) \\ \frac{\partial}{\partial \gamma} \sum_{i=1}^n Y_i(\boldsymbol{x}_i^T \boldsymbol{\beta} + \gamma g_i) - \ln(1+\exp(\boldsymbol{x}_i^T \boldsymbol{\beta} + \gamma g_i)) \end{pmatrix} \\
&= \begin{pmatrix} X^T \boldsymbol{Y} - X^T \boldsymbol{\mu} \\ \sum_{i=1}^n Y_i g_i - \mu_i g_i \end{pmatrix} = \begin{pmatrix} X^T(\boldsymbol{Y} - \boldsymbol{\mu}) \\ \boldsymbol{g}^T(\boldsymbol{Y} - \boldsymbol{\mu}) \end{pmatrix}.
\end{aligned} \tag{11}$$

Observe that:

$$\mathcal{I}_{\gamma\gamma} = F_{\gamma\gamma} = -\frac{\partial^2}{\partial \gamma^2} \ell(\boldsymbol{\theta} \mid \boldsymbol{Y}) = \sum_{i=1}^n \mu_i(1-\mu_i) g_i^2 = \boldsymbol{g}^T W \boldsymbol{g}^T,$$

where W is defined as the $n \times n$ diagonal matrix with $W_{ii} = \mu_i(1-\mu_i)$, and that:

$$-\frac{\partial^2}{\partial \beta_j \partial \gamma} \ell(\boldsymbol{\theta} \mid \boldsymbol{Y}) = \sum_{i=1}^{n} \mu_i(1-\mu_i) x_{i,j} g_i,$$

with $x_{i,j}$ the jth element of the covariate vector $\boldsymbol{x}_i$, and β_j the corresponding parameter. We can therefore write, $\mathcal{I}_{\boldsymbol{\beta}\gamma} = X^T W \boldsymbol{g}$ and $\mathcal{I}_{\gamma\boldsymbol{\beta}} = \mathcal{I}_{\boldsymbol{\beta}\gamma}^T = \boldsymbol{g}^T W X$. Finally, we have that:

$$-\frac{\partial^2}{\partial \beta_j \partial \beta_k} \ell(\boldsymbol{\theta} \mid \boldsymbol{Y}) = \sum_{i=1}^{n} \mu_i(1-\mu_i) x_{i,j} x_{i,k},$$

which means that we can write $\mathcal{I}_{\boldsymbol{\beta}\boldsymbol{\beta}} = X^T W X$. The expected Fisher information, equal to the observed Fisher information, for the logistic regression model is therefore given by

$$F(\boldsymbol{\theta}) = \begin{pmatrix} X^T W X & X^T W \boldsymbol{g} \\ \boldsymbol{g}^T W X & \boldsymbol{g}^T W \boldsymbol{g}^T \end{pmatrix}. \tag{12}$$

Note that the Fischer information indeed is parameter-dependent through the mean vector $\boldsymbol{\mu}$.

We do not know what $\boldsymbol{\mu}$ is. At the same time, our interest is in the hypothesis given in 10. In fact, if we assume the null hypothesis to be true, $\gamma = 0$, the maximum likelihood estimate of $\boldsymbol{\mu}$, denoted $\hat{\boldsymbol{\mu}}$, is by definition given by the equation:

$$X^T(\boldsymbol{Y} - \boldsymbol{\mu}) = \boldsymbol{U}_{\boldsymbol{\beta}} = \mathbf{0}.$$

From (7), we have asymptotically, as $n \to \infty$, under the null hypothesis that:

$$\boldsymbol{U}_\gamma(\boldsymbol{\mu}) \mid (\boldsymbol{U}_{\boldsymbol{\beta}} = \mathbf{0}) \xrightarrow{d} N\left(\mathbf{0}, \boldsymbol{g}^T W \boldsymbol{g}^T - \boldsymbol{g}^T W X (X^T W X)^{-1} X^T W \boldsymbol{g}\right),$$

where $\boldsymbol{U}_\gamma(\hat{\boldsymbol{\mu}}) = \boldsymbol{g}^T(\boldsymbol{Y} - \hat{\boldsymbol{\mu}})$ is the score test statistic in this case.

4.2 Multiple testing, and application in GWAS

In cases where there are a large number of hypothesis tests, such as in GWAS, the probability of committing a Type I error, that is rejecting the null hypothesis when it is indeed true, increases the more tests there are. As a simple example, assume one has s independent tests, each having a probability of type I error equal to ϵ. The probability of falsely rejecting the null hypothesis *at least* once is called the *family-wise error rate* (FWER). The FWER is then given by:

$$FWER = 1 - (1-\epsilon)^s.$$

As $(1-\epsilon) \in (0,1)$, $(1-\epsilon)^s \to 0$ as $s \to \infty$, and so the FWER approaches to have probability one of falsely rejecting the null hypothesis at least once. Therefore, in such a setting one would like to *control* the FWER to be less than some value α.

Assume we have s tests, each with the corresponding statistic T_j, $j = 1, \ldots, s$. Let $p(T_j)$ denote the corresponding *p-value*, a random variable with $0 \leq p(t_j) \leq 1$, with a small p-value indicating the null hypothesis to be unlikely, and a rule that the null hypothesis is rejected when $p(t_j) \leq \lambda$. A p-value is *valid* if $P_{\boldsymbol{\theta}}(T_j \leq \lambda) \leq \lambda$ for every λ and every $\boldsymbol{\theta} \in \boldsymbol{\Theta}_0$, where $\boldsymbol{\Theta}_0$ is the parameter set where the null hypothesis is true. The question is what λ should be in order to control the FWER to be less than some α. One solution is to apply the *Bonferroni correction.* Let $s_0 \leq s$ be the number of tests where the null hypothesis is true. Assume the p-value is valid. Then, by Boole's inequality:

$$\text{FWER} = P\left(\bigcup_{j=1}^{s_0} (p(T_j) \leq \lambda)\right) \leq \sum_{j=1}^{s_0} P(p(T_j) \leq \lambda) \leq s_0 \lambda.$$

Hence, if we set $\lambda = \alpha/s$, then FWER $\leq \frac{s_0}{s}\alpha \leq \alpha$. Note that this result applies no matter whether the tests are independent or not.

In a GWAS, a standard value is to set $\lambda = 5 \times 10^{-8}$, being small precisely to control the number of false rejections. In fact, it can be seen as to originate from a Bonferroni correction in order to control the

FWER at level $\alpha = 0.05$ based on the *effective* number of tests along the whole genome, corresponding to the total number of independent variants along the genome, estimated to be around 10^6 (Dudbridge & Gusnanto, 2008). The FWER is seen as a more rigorous method in order to control the Type I error rate, at the cost of a larger Type II error rate (false negative rate), than other procedures that aim to control the Type I error rate (Goeman & Solari, 2014).

4.3 Covariates and confounders

The purpose of the covariates in a GWAS is often to increase the *power* of the statistical tests, meaning increasing the probability of rejecting the null hypothesis, when the null hypothesis is false. Typical examples of these are adding age and sex. However, there are other reasons to include covariates, such as to avoid *confounding*, a situation in which a *confounder* causes false associations. When inferring the association between a covariate of interest and a response, A covariate is defined as a confounder if it directly influences both the covariate of interest (in our case the SNP), as well as the response. In GWAS, an important confounder is due to *population stratification*, the fact that there are differences in allele frequencies (MAFs), of the same SNPs, between subpopulations in a population, for instance due to the physical distance between them. It turns out that a way to correct for this is to include principal components based on the genotype data (Price et al., 2006).

4.4 When the individuals are related

So far we have assumed the individuals to be unrelated, and the statistical results assumed the individuals to be independent. Typically, the set of individuals in a GWAS are not unrelated. In practice, to reduce the violation of the assumption of independence, one would first need to reduce the total number of individuals to a set of individuals with a sufficiently small degree of mutual relatedness. However, this will reduce the power of the statistical tests. Therefore, the most popular method in GWAS, taking into account relatedness, is to use *mixed-effects* models. See for instance Loh et al. (2015) and Zhou et al. (2018). In this thesis, we will assume the individuals to be unrelated.

5 Saddlepoint approximations

Saddlepoint approximation was first introduced in Daniels (1954), and is a way to estimate the probability distribution of a random variable. It is based on the so-called *cumulant generating function* (CGF), denoted $K(s)$, for a real value $s \in (a, b)$ which is closely related to the *moment generating function* (MGF), denoted $M(s)$, defined for a random variable X as:

$$K(s) = \ln M(s) = E(e^{sX}). \tag{13}$$

If we first assume the random variable X to be continuous, the saddlepoint approximation of the probability density of X, denoted $\hat{f}(x)$, is given by:

$$\hat{f}(x) = \frac{1}{\sqrt{2\pi K''(\hat{s})}} \exp(K(\hat{s}) - \hat{s}x), \tag{14}$$

where $K''(\hat{s})$ is the second derivative of the CGF at the *saddlepoint* $\hat{s}$ which satisfies:

$$K'(\hat{s}) = x, \tag{15}$$

namely that the first derivative of the CGF at the saddlepoint $\hat{s} \in (a, b)$ is equal to the observed valued x. The approximation is achievable for all *interior points* x in which $f(x) > 0$, that is within the *support*, χ, of the probability distribution of f, excluding the boundaries of the support. We let $\mathcal{I}_\chi$ denote the interior of the support of f. Note that even though Expression (14) approximates the density f, it is not itself a density since $\int_\chi \hat{f}(x) \neq 1$. For the derivation of the saddlepoint approximation, we refer to Butler (2007), Chapter 2.

An approximation of the corresponding *cumulative distribution function*, $F(x) = P(X \leq x)$, first introduced in Lugannani and Rice (1980), where $E(X) = \mu$ is given by:

$$\hat{F}(x) = \begin{cases} \Phi(w) + \phi(w)(\frac{1}{w} - \frac{1}{v}), & \text{if } x \neq \mu \\ \frac{1}{2} + \frac{K'''(0)}{6\sqrt{2\pi}K''(0)^{3/2}}, & \text{if } x = \mu, \end{cases} \tag{16}$$

where

$$v = \hat{s}\sqrt{K''(\hat{s})} \qquad w = \operatorname{sgn}(\hat{s})\sqrt{2(\hat{s}x - K(\hat{s}))}, \tag{17}$$

and the functions $\phi()$ and $\Phi()$ denote the standard normal density and cumulative distribution function respectively. When $x \to \mu$, the top expression approaches the bottom expression in (16), and so the entire expression is continuous.

If the random variable X has a discrete distribution, the saddlepoint approximation of the probability mass function, denoted $\hat{p}(x)$, has the exact same expression as in (14):

$$\hat{p}(x) = \frac{1}{\sqrt{2\pi K''(\hat{s})}} \exp(K(\hat{s}) - \hat{s}x), \tag{18}$$

with the saddlepoint $\hat{s}$ defined as in (15). The approximation is meaningful for all interior points x in which $p(x) > 0$, even though the approximation can be computed for any y in which $x_1 < y < x_2$, where x_1 and x_2 are two neighbouring values in $\mathcal{I}_\chi$. Again, the approximation is not achievable at the boundary of the support.

For discrete distributions, Daniels (1987) suggested two so-called *continuity-corrected* saddlepoint approximations to the CDF applied on so-called *lattice distributions.* A lattice distribution has a support of regular grid points with equal distance between neighbouring values. Formally, a discrete random variable has a lattice distribution if the support of the distribution, χ, is on the δ-lattice $\{a, a+\delta, a+2\delta, \ldots\}$ for some real value a and positive real value $\delta \neq 0$. In this thesis, the focus is on one of these continuity-corrections, namely the *second* continuity correction (see Butler (2007), Chapter 1 for the first as well as a third continuity correction). In the rest of this thesis, we will restrict ourselves to discrete random variables with support on the *integer* lattice, meaning $\delta = 1$. Then, the *survival function* defined as $S(x) = P(X \geq x)$ is approximated as:

$$\hat{S}(x) = \begin{cases} 1 - \Phi(\tilde{w}) - \phi(\tilde{w})(\frac{1}{\tilde{w}} - \frac{1}{\tilde{v}}), & \text{if } \ x - \frac{1}{2} \neq \mu \\ \frac{1}{2} - \frac{K'''(0)}{6\sqrt{2\pi}K''(0)^{3/2}}, & \text{if } \ x - \frac{1}{2} = \mu, \end{cases} \tag{19}$$

where

$$\tilde{v} = 2\sinh\left(\frac{\tilde{s}}{2}\right)\sqrt{K''(\tilde{s})} \qquad \tilde{w} = \operatorname{sgn}(\tilde{s})\sqrt{2(\tilde{s}(x - \frac{1}{2}) - K(\tilde{s}))}, \tag{20}$$

where the saddlepoint is denoted $\tilde{s}$ (to make clear the difference between the saddlepoint $\hat{s}$ in the continuous setting), and satisfies:

$$K'(\tilde{s}) = x - \frac{1}{2}. \tag{21}$$

From this result, the corresponding cumulative distribution function can be directly approximated, and is given by $\hat{F}(x) = 1 - \hat{S}(x + 1)$.

5.1 Multivariate distributions

The saddlepoint approximation can be generalized to multivariate distributions. Consider the random vector $\boldsymbol{X}$ of size m, and assume for now it has a continuous distribution, f. Then the corresponding saddlepoint approximation of the multivariate distribution, in the interior of the support of f, is given as:

$$\hat{f}(\boldsymbol{x}) = \frac{1}{(2\pi)^{m/2} \mid H(\hat{\boldsymbol{s}}) \mid^{1/2}} \exp(K(\hat{\boldsymbol{s}}) - \hat{\boldsymbol{s}}^T\boldsymbol{x}), \tag{22}$$

where $K(\hat{\boldsymbol{s}})$ is the CGF of the random vector $\boldsymbol{X}$, $\mid H(\hat{\boldsymbol{s}}) \mid$ is the determinant of the $m \times m$ *Hessian* of the CGF, while the saddlepoint (vector) $\hat{\boldsymbol{s}}$ must satisfy:

$$\nabla K(\hat{\boldsymbol{s}}) = \boldsymbol{x}, \tag{23}$$

where $\nabla K(\hat{\boldsymbol{s}})$ is the *gradient* of the CGF. The saddlepoint approximation of a multivariate discrete random vector X, denoted $\hat{p}(x)$, has the same expression as $\hat{f}(\boldsymbol{x})$, but again only meaningful for interior of the support of p.

5.2 Conditional distributions

Given the random vector $(\boldsymbol{X}, \boldsymbol{Y})$ with $\boldsymbol{X}$ of size m_x and $\boldsymbol{Y}$ of size m_y with $m = m_x + m_y$. Assume for ease of notation that the vector is continuous. The conditional probability density of $\boldsymbol{Y}$ given that $\boldsymbol{X} = \boldsymbol{x}$ is defined as:

$$f(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{f(\boldsymbol{x}, \boldsymbol{y})}{f(\boldsymbol{x})} \quad (\boldsymbol{x}, \boldsymbol{y}) \in \chi. \tag{24}$$

A natural saddlepoint approximation of such conditional distributions, restricted to the interior of the support, can be achieved through the saddlepoint approximation of multivariate distributions:

$$\hat{f}(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{\hat{f}(\boldsymbol{x}, \boldsymbol{y})}{\hat{f}(\boldsymbol{x})} \quad (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{I}_{\mathcal{X}}. \tag{25}$$

This is called *double saddlepoint approximation* since we do saddlepoint approximation on two distributions. In fact, by using the results in Section 1.1, one can show that the double saddlepoint density is given by:

$$\hat{f}(\boldsymbol{y}|\boldsymbol{x}) = (2\pi)^{-m_y/2} \left\{ \frac{\mid H(\hat{\boldsymbol{s}}, \hat{\boldsymbol{t}}) \mid}{\mid H(\hat{\boldsymbol{s}}_0) \mid} \right\}^{-1/2} \times \exp\left(\left(K(\hat{\boldsymbol{s}}, \hat{\boldsymbol{t}}) - \hat{\boldsymbol{s}}^T \boldsymbol{x} - \hat{\boldsymbol{t}}^T \boldsymbol{y}\right) - \left(K(\hat{\boldsymbol{s}}_0) - \hat{\boldsymbol{s}}_0^T \boldsymbol{x}\right)\right), \tag{26}$$

where $\mid H(\hat{\boldsymbol{s}}, \hat{\boldsymbol{t}}) \mid$ is the determinant of the Hessian of the CGF, $K(\boldsymbol{s}, \boldsymbol{t})$, of the joint distribution $f(\boldsymbol{x}, \boldsymbol{y})$, evaluated at the saddlepoint $\begin{pmatrix}\hat{\boldsymbol{s}}^T & \hat{\boldsymbol{t}}^T\end{pmatrix}^{\mathrm{T}}$ which satisfies

$$\nabla_{\boldsymbol{s},\boldsymbol{t}} K(\hat{\boldsymbol{s}}, \hat{\boldsymbol{t}}) = \begin{pmatrix}\boldsymbol{x}^T & \boldsymbol{y}^T\end{pmatrix}^{\mathrm{T}}, \tag{27}$$

with $\nabla_{\boldsymbol{s},\boldsymbol{t}} K(\boldsymbol{s}, \boldsymbol{t})$ the gradient of the CGF $K(\boldsymbol{s}, \boldsymbol{t})$ with respect to $\boldsymbol{s}$ and $\boldsymbol{t}$ (in that order), where $\boldsymbol{s}$ and $\boldsymbol{t}$ are associated with $\boldsymbol{x}$ and $\boldsymbol{y}$ respectively. The expression $\mid H(\hat{\boldsymbol{s}}_0) \mid$ is the determinant of the Hessian with respect to the CGF, $K(\boldsymbol{s})$, of the marginal distribution of $\boldsymbol{X}$, $f(\boldsymbol{x})$, evaluated at the saddlepoint $\hat{\boldsymbol{s}}_0$ that satisfies

$$\nabla_{\boldsymbol{s}} K(\hat{\boldsymbol{s}}_0) = \boldsymbol{x}, \tag{28}$$

with $\nabla_{\boldsymbol{s}} K(\boldsymbol{s})$ the gradient of $K(\boldsymbol{s})$. The results can be generalized to discrete distributions simply by replacing the symbol f with p above.

The approximation of the cumulative distribution function conditional on $\boldsymbol{x}$, denoted $\hat{F}(y \mid \boldsymbol{x})$, in the case where $m_y = 1$ was first derived in Skovgaard (1987). We restrict ourselves to all $y \neq E(Y \mid x)$:

$$\hat{F}(y \mid \boldsymbol{x}) = \Phi(w) + \phi(w) \left(\frac{1}{w} - \frac{1}{v} \right) \tag{29}$$

with

$$v = \hat{t} \sqrt{\frac{\mid H(\hat{\boldsymbol{s}}, \hat{t}) \mid}{\mid H(\hat{\boldsymbol{s}}_0) \mid}} \qquad w = \operatorname{sgn}(\hat{t}) \sqrt{2\left(\left(K(\hat{\boldsymbol{s}}_0) - \hat{\boldsymbol{s}}_0^T \boldsymbol{x}\right) - \left(K(\hat{\boldsymbol{s}}, \hat{t}) - \hat{\boldsymbol{s}}^T \boldsymbol{x} - \hat{t} y\right)\right)}. \tag{30}$$

5.2.1 Integer lattice distributions

As for the univariate case, continuity corrections for the cumulative distribution function of $p(y \mid \boldsymbol{x})$ are available when Y has an integer lattice distribution. Of notice is that this is achievable no matter the type of support of $\boldsymbol{X}$ (lattice or continuous, or even a mix). The second continuity correction of the survival function, $\hat{S}(y) = P(Y \geq y \mid \boldsymbol{X} = \boldsymbol{x})$, is given by

$$\hat{S}(y) = 1 - \Phi(\tilde{w}) - \phi(\tilde{w})(\frac{1}{\tilde{w}} - \frac{1}{\tilde{v}}) \qquad y \neq E[Y \mid \boldsymbol{X} = \boldsymbol{x}], \tag{31}$$

with

$$\tilde{v} = 2\sinh\left(\frac{\tilde{t}}{2}\right)\sqrt{\frac{\mid H(\tilde{\boldsymbol{s}},\tilde{t})\mid}{\mid H(\tilde{\boldsymbol{s}}_0)\mid}} \qquad \tilde{w} = \operatorname{sgn}(\tilde{t})\sqrt{2\left(\left(K\left(\tilde{\boldsymbol{s}}_0\right) - \tilde{\boldsymbol{s}}_0^T\boldsymbol{x}\right) - \left(K\left(\tilde{\boldsymbol{s}},\tilde{t}\right) - \tilde{\boldsymbol{s}}^T\boldsymbol{x} - \tilde{t}\left(y - \frac{1}{2}\right)\right)\right)}, \tag{32}$$

where the saddlepoint $\begin{pmatrix}\tilde{\boldsymbol{s}}^T & \tilde{t}^T\end{pmatrix}^{\mathrm{T}}$ must satisfy

$$\nabla_{\boldsymbol{s},t}K(\tilde{\boldsymbol{s}},\tilde{t}) = \begin{pmatrix}\boldsymbol{x}^T & y - \frac{1}{2}\end{pmatrix}^{\mathrm{T}}, \tag{33}$$

while the saddlepoint $\tilde{\boldsymbol{s}}_0$ must satisfy

$$\nabla_{\boldsymbol{s}}K(\tilde{\boldsymbol{s}}_0) = \boldsymbol{x}. \tag{34}$$

5.2.2 Alternative approximation to the CDF

An alternative to the saddlepoint approximation of the CDF given in (16), was introduced by Barndorff-Nielsen (1990) and is given by:

$$\hat{F}(x) = \Phi\left(w + \frac{1}{w}\log\left(\frac{v}{w}\right)\right) \qquad x \neq \mu, \tag{35}$$

with v and w unchanged and defined as in (17). The results given above, using (16), is equally valid when using (35). For instance, the conditional survival function for integer lattice distributions using (35) is simply given by:

$$\hat{S}(y) = 1 - \Phi\left(\tilde{w} + \frac{1}{\tilde{w}}\log\left(\frac{\tilde{v}}{\tilde{w}}\right)\right) \qquad y \neq E[Y \mid \boldsymbol{X} = \boldsymbol{x}],$$

with $\tilde{v}$ and $\tilde{w}$ defined as in (32).

6 Statistical learning and inference

Given a model matrix $X_{n\times p}$ including n samples $\boldsymbol{x}_i$ each including the observed value from p covariates, from hereon denoted features, as well as corresponding univariate *response* values $\mathbf{y}$ for each sample. Assume there exists some unknown data generating process where a response value, Y, is generated according to a probability distribution depending on the corresponding observed feature values $\mathbf{x}$. We denote a machine learning model to be a function, $\hat{y}(\mathbf{x})$, that approximates the unknown data generating process. Let the *loss function*, $\ell(y,\hat{y}(\mathbf{x}))$, denote some measure for the distance between the observed response value, y, and the corresponding predicted response value $\hat{y}(\mathbf{x}_i)$. The machine learning model is constructed by searching for the model that minimizes the expected loss per sample. Let $\hat{y}^*(\mathbf{x})$ denote this ideal function. Then

$$\hat{y}^*(\mathbf{x}) = \arg\min_{\hat{y}} E_{\mathbf{X},Y}[\ell(Y,\hat{y}(\mathbf{X}))].$$

The procedure is to generate a model $\hat{y}(\mathbf{x})$ that is as close as possible to $\hat{y}^*(\mathbf{x})$ by using the data $X_{n\times p}$ and $\mathbf{y}$. Broadly speaking, statistical learning is the theory and methods behind the process that leads to the construction of the model $\hat{y}(\mathbf{x})$, as well as how to assess the quality of the model (Hastie et al., 2009). Statistical inference refers to the theory behind how to draw conclusions about some unknown parameter or measure, in which one is interested in estimating (Casella & Berger, 2001).

6.1 Model assessment

Having constructed the model $\hat{y}(\mathbf{x})$, the question is how well it reflects the unknown underlying generating process, and therefore how well it *generalizes* to new data. A natural way to measure this is to consider the *expected prediction error*, $\mathrm{Err}(x_0) = E_{\mathbf{X}_0,Y_0}\left[\ell(Y_0,\hat{y}(\mathbf{X}_0))\right]$, based on a *new* sample $\mathbf{x}_0$ and corresponding response value y_0 generated from the same underlying probability distribution.

Consider the data generating process $Y = f(\boldsymbol{X}) + \epsilon$, with $E(\epsilon) = 0$ and $\mathrm{Var}(\epsilon) = \sigma_\epsilon^2$. Let Y be continuous, and therefore can take any real-valued number. We apply the loss, $\ell(y,\hat{y}(\mathbf{x})) = (y - \hat{y}\left(\mathbf{x}\right))^2$, the squared distance between the true response value, and the response value predicted by the model. Then one can show that (Hastie et al., 2009)

$$\mathrm{Err}(\mathbf{x}_0) = E_{\mathbf{X}_0,Y_0}\left[(Y_0 - \hat{y}(\mathbf{X}_0))^2\right] = \sigma_\epsilon^2 + [E[\hat{y}(\mathbf{X}_0)] - f(\mathbf{X}_0)]^2 + \mathrm{Var}(\hat{y}(\mathbf{X}_0)). \tag{36}$$

The first term in (36) is the variance of the irreducible error which we can not control. The second term is the *bias* squared, where the bias is the expected difference between the output of the model and the true output. The last term is simply the variance of the model output. When the variance of the model is large, this can be as a result that the model imitates the training data too much, not accounting for random noise. This is called *overfitting*. On the other hand, if the bias is too large, this can be as a result that the model has not captured important relationships between the features and the response. We call this *underfitting*. The bias-variance tradeoff is the situation where the best model will need to have neither too large variance nor too large bias, a tradeoff between finding important relationships between the features and the model, but at the same time not falsely modelling random noise. The idea of bias-variance tradeoff can be generalized for other loss functions than the squared error.

6.1.1 Training data, validation data and test data

To quantify how well a model generalizes to new data, and to observe potential underfitting or overfitting, the normal procedure is to disjointly split data in *training data*, *validation data* and *test data*. The training data is used to fit the model. Validation data, never observed during training, can be used to compare several models, or used to evaluate the progression of the model during model fitting. The test data, never used during training or validation, is used to measure how well the constructed model generalizes to new data.

6.1.2 Estimation of test error and expected prediction error

We denote the *test error*, $\mathrm{Err}_{\mathcal{T}} = E_{\mathbf{X},Y}[\ell(Y, \hat{y}(\mathbf{X}))|\mathcal{T}]$ as the expected prediction error conditioned on the training data, $\mathcal{T}$. It turns out that this measure is difficult to estimate (Hastie et al., 2009). However, the expected prediction error, $\mathrm{Err} = E_{\mathcal{T}}[\mathrm{Err}_{\mathcal{T}}]$, including the randomness in the training data, can be estimated by *cross-validation*. In cross-validation, the data is randomly split in K disjoint sets. For each iteration, $K-1$ sets are used to train a model, while the last set is used as test data. Let $\hat{y}^{k(i)}(\mathbf{x}_i)$ denote the prediction with respect to the pair $(\mathbf{x}_i, y_i)$ based on the model constructed by the training data $k(i)$ that does not include $(\mathbf{x}_i, y_i)$. Then the estimate of the expected prediction error is given by:

$$\widehat{\mathrm{Err}}(\hat{y}) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \hat{y}^{k(i)}(\mathbf{x}_i)).$$

6.2 Bootstrapping

By having an *estimator* of some unknown parameter of interest, the uncertainty in the corresponding *estimate* is often evaluated by making assumptions about the underlying probability distribution of the estimator, for instance based on some parametric model, as well as by using *maximum likelihood theory* (Casella and Berger (2001)). However, there may be circumstances where it is difficult to come up with reasonable assumptions about the probability distribution of the estimator. In this case, we may use *bootstrapping* (Efron & Tibshirani, 1994) instead in order to infer the distribution of the estimator without making any assumptions of the underlying probability distribution. The general procedure in bootstrapping is to iteratively resample the data at hand *with replacement*, and for each iteration, compute the estimate based on this bootstrap sample. Specifically, consider we have the data $(\mathbf{z}_1, \dots, \mathbf{z}_n)$ of size n. Then a bootstrap sample $(\mathbf{z}_1^*, \dots, \mathbf{z}_n^*)$ of size n is generated by sampling from the *empirical distribution* that assigns equal probability of sampling each data point each time equal to $1/n$. For some estimator $T(\mathbf{Z}_1, \dots, \mathbf{Z}_n)$, the corresponding estimate for this bootstrap sample is then given by $T(\mathbf{z}_1^*, \dots, \mathbf{z}_n^*)$. By having B bootstrap iterations, we get B observations from the estimator.

By resampling the data with replacement, we imitate sampling of data generated from the true underlying probability distribution, and hence we may imitate the true probability distribution of the corresponding estimator after sufficiently many iterations. From the estimated probability distribution,

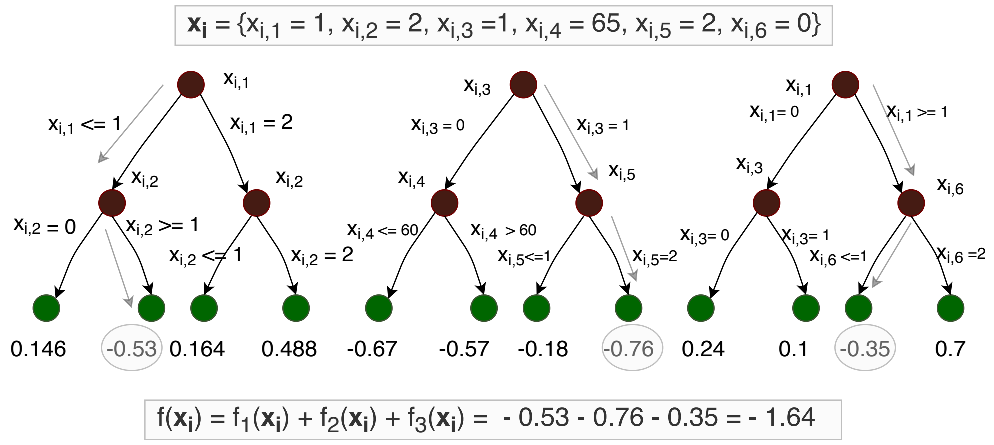


Figure 1: An example with three constructed regression trees with six features $x_{i,1}$ to $x_{i,6}$ used as splitting points at each branch, and leaf node values. Also shown is the computation of $f(\mathbf{x}_i)$ given an example of feature values $\mathbf{x}_i$. The structure of the trees opens the possibility to explore interactions since a path from a root node to a leaf node denotes a combination of feature values (**copy from Paper 2 published in BMC Bioinformatics**).

we may also construct confidence intervals. The accuracy in the estimated probability distribution of the estimator, and the corresponding confidence interval, naturally depends on the size of the data, and the number of bootstrap iterations (Efron & Tibshirani, 1994).

7 Tree ensemble models

Tree ensemble models is a type of machine learning model which is a member of the class of *ensemble models*, sometimes also denoted *additive models*. What all ensemble models have in common is that they include several *base learners*. Given a model $\hat{y}(\mathbf{x}_i)$, J base learners $f_j(\mathbf{x}_i)$, and some transformation function $g(\cdot)$, then the ensemble model is written as

$$\hat{y}(\mathbf{x}_i) = g\left(\sum_{j=1}^{J} f_j(\mathbf{x}_i)\right).$$

What exactly the base learners look like, and whether they are of the same structure is up to the user. For regression tree ensemble models, *all* base learners are so-called *regression trees*[1]. A regression tree is a function including *nodes*, *leaves* and *branches*. In each node, there is a binary split creating two branches. The output of the function with feature values $\mathbf{x}_i$, is given by starting at the *root node*. At each split, there is a *splitting point* equal to one of the features, and a rule including a splitting value deciding which branch to move along depending on the feature value of this feature in $\mathbf{x}_i$. At a leaf node, there is some leaf value, continuous and real valued, which will be the output of the regression tree. In the ensemble model, the regression tree, $f_j(\mathbf{x}_i)$, may be multiplied by some constant, $f_j(\mathbf{x}_i) = \eta f_j^*(\mathbf{x}_i)$, with $f_j^*(\mathbf{x}_i)$ the raw regression tree. The output from the regression trees is simply the additive output of each single regression tree. See Figure 1 for an example with three regression trees of *depth* two (two *generations* after the root node), with $\eta = 1$ for each regression tree, and the corresponding tree ensemble output. The regression trees are typically *symmetric*, meaning that there are leaf nodes only at the last *level* of the tree. But there may also be leaf nodes closer to the root node.

Whether the exact structure of each base learner is decided before fitting the model, or if the structure of each base learner is built during model fitting is in principle up to the user. For instance for tree

[1] The base learners in a tree ensemble model can also be a *classification tree*, but for brevity we have ommited this type of tree ensemble model.

ensemble models: Whether the structure of the regression tree is predetermined, and only the leaf values are decided during model fitting, or if each regression tree is built during the model fitting, is up to the user. Naturally, one would like to be as least biased and restrictive as possible before fitting the model, and so the last option is preferred, namely constructing the base learners *during* model fitting. What is typically done, is that each base learner is constructed sequentially, one by one, and each base learner is constructed by minimizing some *loss function*, in which the transformation function $g()$ is naturally decided. For instance, consider after model fitting a resulting tree ensemble model $\hat{y}(\mathbf{x}_i) = g\left(\sum_{\tau=1}^{T} f_\tau(\mathbf{x}_i)\right)$ including T regression trees. Let $X_{n\times p}$ denote the matrix of n feature samples and p features, with observed feature values $\mathbf{x}_i$ at each row, and let $\mathbf{y}$ denote the vector of corresponding response values. For a linear regression problem with continuous response, a natural loss function is the mean square error for which the identity function, $g(x) = x$ is a natural choice. In this case, the loss function, $L(X_{n\times p}, \mathbf{y}, \hat{\mathbf{y}})$, is given by

$$L(X_{n\times p}, \mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}(\mathbf{x}_i))^2 = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \left(\sum_{\tau=1}^{T} f_\tau\left(\mathbf{x}_i\right)\right)\right)^2.$$

The purpose of the model fitting is to search for the model $\hat{y}(\mathbf{x}_i)$ that minimizes the loss function, which is an estimate of the expected squared error, $E[\ell(y_i, \hat{y}(\mathbf{x}_i)]$, where $\ell(y_i, \hat{y}(\mathbf{x}_i)) = (y_i - \hat{y}(\mathbf{x}_i))^2$ in this case. For a classification problem with binary response values, that is y_i is either zero or one, a typical loss function is the estimate of the binary cross entropy

$$L(X_{n\times p}, \mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n}\sum_{i=1}^{n}\ell(y_i, \hat{y}(\mathbf{x}_i)),$$

with $\ell(y_i, \hat{y}(\mathbf{x}_i)) = -y_i \log(\hat{y}(\mathbf{x}_i)) - (1-y_i)\log(1-\hat{y}(\mathbf{x}_i))$. In this particular case, $\hat{y}(\mathbf{x}_i)$ must indicate a *probability*, hence equal to a number between zero and one. A natural transformation function, applied in logistic regression models, is to use the logit function $g(x) = \text{logit}(x)$ in which

$$\hat{y}(\mathbf{x}_i) = \frac{1}{1 + e^{-\sum_{\tau=1}^{T} f_\tau(\mathbf{x}_i)}},$$

which will make sure that $\hat{y}(\mathbf{x}_i) \in (0,1)$[2].

7.1 Bagging and column sampling

Having training data $X_{n\times p}$ and $\mathbf{y}$, a natural procedure for ensemble models would be to fit the base learner using the same training data each time. However, this results in an expected *prediction error* to become large due to overfitting (Hastie et al. (2009), Chapter 10). We present two strategies on how to reduce overfitting in ensemble models.

7.1.1 Bagging

The idea of *bagging* is, for each base learner, to first make *bootstrap* samples, namely resampling the training data *with* replacement, and then fit the base learner. The output of the ensemble model, consisting of J base learners, is then the average of all base learners ($\eta = 1/J$). In that way, the bootstrap samples used for each base learner resembles the observed samples coming from the underlying probability distribution function of the data. Let us assume that each base learner estimator is identically distributed with some variance σ^2. Then averaging the base learners will not reduce the bias. However, assume ρ to be the correlation between any pair of base learner estimators. Then one can show that the variance, $\text{Var}(\hat{y}(\mathbf{X}_0))$, given B base learners is given by

$$\text{Var}(\hat{y}(\mathbf{X}_0)) = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2.$$

[2] This type of transformation makes sense for regression tree ensemble models using *boosting*, a model fitting procedure to be introduced later. For tree ensemble models using classification trees, we refer to Hastie et al. (2009), Chapter 9.

In other words, the larger the number of base learners in a bagging process, the smaller the second expression will be, and so the ensemble model will have a reduced variance, and so performing better with respect to the expected prediction error. Bagging of regression trees turned out to be particularly useful, as regression trees with sufficiently depth are low-biased, yet with large variance, even for complex relationships such as interactions.

7.1.2 Column sampling

Even though the bagging process reduces the variance of the model, we still have the correlation, ρ, between the base learners. An idea to reduce this correlation is to moderate the splitting decision when fitting each base learner, a strategy that is applied in random forest models (Breiman, 2001). Specifically, for each split, randomly sample $m < p$ features, and use only these features to make a splitting decision (choose which feature to split on, and by what value). By doing this, the features are more spread along all trees, and so two randomly chosen trees will have less features in common, and correlate less. This procedure turned out to be particularly successful when using trees as base learners. In general, this strategy with the aim of reducing the model variance, is often denoted *column sampling*, as we in principle randomly sample along the columns of the data matrix $X_{n\times p}$ to choose which m features to use in each splitting decision.

7.2 Boosting

So far, for bagging and for random forests, each regression tree is constructed independently of the others in such a way that the information from the previous constructed regression trees are not used to construct a new regression tree. In addition, for high-dimensional data where only a small fraction of the features are important, random forest models are expected to perform poorly, as the features are randomly selected in each regression tree. Notice also that each base learner is considered equally important because $\eta = 1/B$ for all the B base learners.

What all *boosting* models have in common is that each base learner is fitted based on the previous fitted base learners. Specifically for tree boosting models: Based on the history of the previously fitted trees, and the corresponding ensemble model created so far, the next tree is constructed with the aim of minimizing the loss function of the updated ensemble model. See examples in Chapter 10 in Hastie et al. (2009). We will now introduce one such tree boosting model that is implemented in the *XGBoost* software (Chen & Guestrin, 2016).

7.2.1 XGBoost

XGBoost, an abbreviation for extreme gradient boosting is a popular software for producing a tree ensemble model of the type $\hat{y}(\mathbf{x}_i)^{(T)} = \sum_{\tau=1}^{T} f_\tau(\mathbf{x}_i)$ consisting of T regression trees. The total loss function after the construction of T trees is given by

$$L(X_{n\times p}, \mathbf{y}, \hat{y}^{(T)}) = \sum_{i=1}^{n} \ell\left(y_i, \hat{y}(\mathbf{x}_i)^{(T)}\right) + \sum_{\tau=1}^{T} \gamma V_\tau + \frac{1}{2}\lambda||\mathbf{v}_\tau||^2,$$

for some first and second differentiable convex loss function per sample $\ell(y_i, \hat{y}(\mathbf{x}_i))$, and where γ and λ are predetermined *regularization* parameters, while $\mathbf{v}_\tau$ and V_τ are the vectors of leaf values and the total number of leaves in regression tree τ, respectively. The boosting procedure is as follows: Given $t-1$ fitted regression trees, and corresponding model $\hat{y}(\mathbf{x}_i)^{(t-1)} = \sum_{\tau=1}^{t-1} f_\tau(\mathbf{x}_i)$, the aim is, given the structure of tree t denoted f_t, to find the leaf values $\mathbf{v}_t^*$ that minimize the updated loss function:

$$\min_{\mathbf{v}_t^*} L(X_{n\times p}, \mathbf{y}, \hat{y}^{(t-1)}, f_t, \boldsymbol{v}_t) = \sum_{i=1}^{n} \ell(y_i, \hat{y}(\mathbf{x}_i)^{(t-1)} + f_t(\mathbf{x}_i)) + \sum_{\tau=1}^{t-1} \gamma V_\tau + \frac{1}{2}\lambda||\mathbf{v}_\tau||^2 + \gamma V_t + \frac{1}{2}\lambda||\mathbf{v}_t||^2.$$

In XGBoost, *Newton boosting* is used to approximate the solution by estimating the loss function as a quadratic function by applying a second order Taylor expansion:

$$\begin{aligned} &L(X_{n\times p},\mathbf{y},\hat{y}^{(t-1)},f_t,\boldsymbol{v}_t) \approx \tilde{L}(X_{n\times p},\mathbf{y},\hat{y}^{(t-1)},f_t,\boldsymbol{v}_t) \\ &= \sum_{i=1}^{n} \ell(y_i,\hat{y}(\mathbf{x}_i)^{(t-1)}) + g_i f_t(\mathbf{x}_i) + h_i f_t(\mathbf{x}_i)^2 + \sum_{\tau=1}^{t-1}\left(\gamma V_\tau + \frac{1}{2}\lambda||\mathbf{v}_\tau||^2\right) + \gamma V_t + \frac{1}{2}\lambda||\mathbf{v}_t||^2, \end{aligned} \tag{37}$$

with $g_i = \frac{\partial}{\partial \hat{y}(\mathbf{x}_i)^{(t-1)}}\ell(y_i,\hat{y}(\mathbf{x}_i)^{(t-1)})$ and $h_i = \frac{1}{2}\frac{\partial^2}{\partial \hat{y}^2(\mathbf{x}_i)^{(t-1)}}\ell(y_i,\hat{y}(\mathbf{x}_i)^{(t-1)})$.

Due to the convexity of $\tilde{L}$, and that each leaf value in tree t can take any real value, the weights $\mathbf{v}_t^*$ that minimizes $\tilde{L}(X_{n\times p},\mathbf{y},\hat{y}^{(t-1)},f_t,\boldsymbol{v}_t)$ is a unique solution of

$$\frac{\partial}{\partial \boldsymbol{v}_t}\tilde{L}(X_{n\times p},\mathbf{y},\hat{y}^{(t-1)},f_t,\boldsymbol{v}_t^*) = \mathbf{0}.$$

Let I_j denote the set of all samples which leads to the same leaf node j in the regression tree t, $I_j = \{\forall \;\; i|f_t(\mathbf{x}_i) = v_{t,j}\}$. Then we may rewrite (37) as

$$\begin{aligned} \tilde{L}(X_{n\times p},\mathbf{y},\hat{y}^{(t-1)},f_t,\boldsymbol{v}_t) &= \sum_{j=1}^{V_t}\left(\left(\sum_{i\in I_j} g_i\right) v_{t,j} + \frac{1}{2}\left(\sum_{i\in I_j} h_i + \lambda\right) v_{t,j}^2\right) \\ &+ \sum_{i=1}^{n}\ell(y_i,\hat{y}(\mathbf{x}_i)^{(t-1)}) + \sum_{\tau=1}^{t-1}\left(\gamma V_\tau + \frac{1}{2}\lambda||\mathbf{v}_\tau||^2\right) + \gamma V_t. \end{aligned} \tag{38}$$

The leaf values $\boldsymbol{v}_t^*$ in the regression tree f_t that minimizes (38) is given by

$$v_{t,j}^* = \frac{-\sum_{i\in I_j} g_i}{\sum_{i\in I_j} h_i + \lambda},$$

with the corresponding updated loss equal to

$$\begin{aligned} \tilde{L}(X_{n\times p},\mathbf{y},\hat{y}^{(t)}) &= -\frac{1}{2}\sum_{j=1}^{V_t}\left(\frac{(\sum_{i\in I_j} g_i)^2}{\sum_{i\in I_j} h_i + \lambda}\right) \\ &+ \sum_{i=1}^{n}\ell(y_i,\hat{y}(\mathbf{x}_i)^{(t-1)}) + \sum_{\tau=1}^{t-1}\left(\gamma V_\tau + \frac{1}{2}\lambda||\mathbf{v}_\tau||^2\right) + \gamma V_t. \end{aligned} \tag{39}$$

However, we do not know what the structure of the tree should be. Instead, a *greedy* approach is developed when constructing each regression tree. One starts at the root node, and investigate every feature and every possible splitting rule, and look at the corresponding reduced loss function in (39). The feature and splitting rule that gives the smallest updated loss function is then used. The algorithm continues to split on the newly created leaf nodes in the same manner. This is however an exhaustive search method, and there exist other splitting rule algorithms that are not exact, however faster and provide good approximations such as the *histogram method* (Alsabti et al., 1998; Jin & Agrawal, 2003; Li et al., 2008) which is implemented in the XGBoost software packages.

7.2.2 Reducing overfitting in boosting models

Using only the training data during model fitting, the model will overfit as the number of trees T increases. One must therefore have a rule that stops the algorithm before it begins to overfit. One solution is to disjointly separate the data in training data, validation data and test data. The training data is used to fit the model. However, for each step of the model fitting procedure, the loss function is applied to the validation data to quantify performance. Note that the validation data is never used to fit the model, and one can therefore reliably evaluate the model at each step. One possible rule is to stop the training of the model when the total loss based on the validation data has not decreased within a given number of consecutive updates. We denote the parameters that control the learning process of the boosting model as *hyperparameters*. For XGBoost, one can use the *early_stopping_rounds* hyperparameter in the software for this purpose. To evaluate how well the fitted model generalizes to new data, this can be assessed by using the test data.

The purpose of the regularization hyperparameters λ and γ is to reduce the variance of the model, by reducing the effect each base learner will have on the full model such as pruning large leaf values and large trees. There are several possible rules to incorporate when creating each tree. For instance, if the proposed splitting according to the algorithm leads to an increased loss, one may decide to stop training this regression tree, and create a new one (using the regression tree from before the splitting). Another option is to limit the maximum depth of each regression tree. Similarly to random forests, one can sample a reduced set of features for each splitting to reduce overfitting as well as reducing the running time.

What turns out to reduce the overfitting even further for boosting models is to randomly choose a subset of samples, without replacement, for each regression tree referred to as *stochastic gradient boosting*, or *subsampling* (Friedman, 2002). This will also reduce the running time in each iteration. Another important hyperparameter in boosting models is the *learning rate*, which is the constant η multiplied by the raw regression tree. By setting this constant to some small value below one, this will also limit the effect each base learner has on the full model. A smaller learning rate will also typically generate a model with a larger number of regression trees.

8 Shapley values

The *Shapley* value was first introduced by Lloyd Shapley (1953) originally to be applied within game theory which involves mathematical modelling of a system in which agents, often denoted players, interact with each other, and where each player makes rational decisions on how to maximize its own *value*, for instance by cooperation with other players (cooperative game). The purpose of the Shapley value is to measure the contribution, or value, for each player in a given cooperative game compared to the total value of the cooperative game. The use of Shapley values is an example of a *payment rule* in a game. Formally, let there be M players in the cooperative game. Let $\mathcal{S}$ be a set of players, such as $\{1,2,3\}$, meaning player one, two and three. We define a *value* function, $v(\mathcal{S})$, which is a measure of the total value or payoff for the players in $\mathcal{S}$. In this setting we define $v(\emptyset) = 0$, with $\emptyset$ the empty set. Let $\mathcal{M}$ denote the set of all players in the game. The Shapley value, ϕ_k, for player k is then defined as:

$$\phi_k = \sum_{\mathcal{S} \subseteq \mathcal{M} \setminus \{k\}} \frac{|\mathcal{S}|!(M - |S| - 1)!}{M!} (v(\mathcal{S} \cup \{k\}) - v(\mathcal{S})). \tag{40}$$

Here, $\mathcal{S} \subseteq \mathcal{M} \setminus \{k\}$ means any achievable subset $\mathcal{S}$, among all 2^M possible subsets, not including player k. The Shapley values have several desirable properties:

Symmetry. *Given players j and k where $v(\mathcal{S} \cup \{j\}) = v(\mathcal{S} \cup \{k\})$ for all $\mathcal{S} \subseteq \mathcal{M} \setminus \{j,k\}$. Then $\phi_j = \phi_k$,*

Dummy player. *For a player j with $v(\mathcal{S} \cup \{j\}) = v(\mathcal{S})$ for all $\mathcal{S} \subseteq \mathcal{M} \setminus \{j\}$, $\phi_j = 0$.*

Linearity. *Given two independent games consisting of the same players, but with different value functions v and w. Then $\phi(v + w) = \phi(v) + \phi(w)$.*

Efficiency. $\sum_{k=1}^{M} \phi_k = v(\mathcal{M}) - v(\phi) = v(\mathcal{M})$.

The symmetry property means that two players with equal payoffs (value) with the same coalitions will have the same Shapley value, and therefore considered equally worthy in the game, and so has the same Shapley value. If a player has no payoff with any coalition, then the player has no worth and the Shapley value equals zero. The linearity property is a direct result of the linearity of sums. The efficiency property is the least intuitive, and deserves a proof. This can be achieved by applying an equivalent definition of the Shapley value:

$$\phi_k = \frac{1}{M!} \sum_{R} \left[v(s_k(R) \cup \{k\}) - v(s_k(R)) \right] , \tag{41}$$

where the sum is over all *orderings* R of the M features, with a total of $M!$ orders. The function $s_k(R)$ maps a given ordering R and a particular feature k to the corresponding subset of features preceding feature k in the specific ordering. For instance, for $\mathcal{M} = \{1, 2, 3\}$, one possible ordering is $R = (2, 3, 1)$ with $s_1(R) = (2, 3)$.

Proof.

$$\begin{aligned}
\sum_{k=1}^{M} \phi_k &= \sum_{k=1}^{M} \frac{1}{M!} \sum_{R} \left[v(s_k(R) \cup \{k\}) - v(s_k(R)) \right] \\
&= \frac{1}{M!} \sum_{R} \sum_{k=1}^{M} \left[v(s_k(R) \cup \{k\}) - v(s_k(R)) \right] \\
&= \frac{1}{M!} \sum_{R} \left(v(\mathcal{M}) - v(\emptyset) \right) \\
&= \frac{1}{M!} M! \left(v(\mathcal{M}) - v(\emptyset) \right) = v(\mathcal{M}) - v(\emptyset),
\end{aligned} \tag{42}$$

since for a specific ordering R and feature k, in the sum $\sum_{k=1}^{M} [v(s_k(R) \cup \{k\}) - v(s_k(R))]$ all terms cancel each other, except $v(\mathcal{M})$ and $v(\emptyset)$. Hence, with $v(\emptyset) = 0$, $\sum_{k=1}^{M} \phi_k = v(\mathcal{M})$. □

As a result, the efficiency property means that the sum of the contribution for each player is equal to the total contribution in the game. Hence, the Shapley values can be compared with each other in such a way that if $\phi_k > \phi_j$, then player k has a larger contribution than player j in the game. In fact, Lloyd Shapley (1953) showed that the *only* payment rule that satisfies the properties of symmetry, dummy player, linearity and efficiency is by the construction of the Shapley values.

8.1 Shapley values in machine learning

With the increased popularity of complex high-dimensional machine learning models such as deep neural networks and ensemble models, there has been an increased interest in developing methods to interpret such models, often called *black-box* models. There are mainly two reasons for why this is important. One reason is to analyse *whether* the predictions of the models make sense. The other reason is to find out *what* the model considers important for each prediction. The idea of using Shapley values to explain what a model considers important was introduced for linear regression models in Lipovetsky and Conklin (2001) using the *coefficient of determination*, R^2, as value function. In more detail, for each subset $\mathcal{S} \subseteq \mathcal{M}$ of all M features/covariates, a linear regression model was fitted based on this subset $\mathcal{S}$, and a resulting value function defined by $v(\mathcal{S}) = R^2_{\mathcal{S}}$ for this subset with $v(\emptyset) = 0$.

For high-dimensional data, the approach in Lipovetsky and Conklin (2001) to fit a model for each subset $\mathcal{S}$, out of all 2^M subsets, can quickly become infeasible as the number of subsets increases exponentially for increasing number of features. In Štrumbelj and Kononenko (2014), the use of Shapley values in regression models was generalized by defining the same value function independently of which type of regression model was analysed, however applied to the same prediction model for each subset. Specifically, the value function, $v_{\mathbf{x}_i, \hat{y}}(\mathcal{S})$, for feature values $\mathbf{x}$ and a fitted regression model $\hat{y}$ is defined as

$$v_{\mathbf{x}, \hat{y}}(\mathcal{S}) = E_{\mathbf{X}_{\overline{\mathcal{S}}}}[\hat{y}(\mathbf{X} | \mathbf{X}_{\mathcal{S}} = \mathbf{x}_{\mathcal{S}})], \tag{43}$$

which means for $\mathcal{S} = \emptyset$ that

$$v_{\mathbf{x}, \hat{y}}(\emptyset) = E_{\mathbf{X}}[\hat{y}(\mathbf{X})].$$

For instance, if we assume all features to take continuous values, then

$$v_{\mathbf{x}, \hat{y}}(\mathcal{S}) = E_{\mathbf{X}}[\hat{y}(\mathbf{X})] = \int_{\mathbf{x}_{\overline{\mathcal{S}}}} \hat{y}(\mathbf{x}_{\mathcal{S}}, \mathbf{x}_{\overline{\mathcal{S}}}) p(\mathbf{X}_{\overline{\mathcal{S}}} = \mathbf{x}_{\overline{\mathcal{S}}} | \mathbf{X}_{\mathcal{S}} = \mathbf{x}_{\mathcal{S}}) d\mathbf{x}_{\overline{\mathcal{S}}}, \tag{44}$$

where $\mathbf{x}_{\mathcal{S}}$ denotes the subset of feature values in the observed vector $\mathbf{x}$ for the features in the subset $\mathcal{S}$, and $p(\mathbf{X}_{\overline{\mathcal{S}}} = \mathbf{x}_{\overline{\mathcal{S}}} | \mathbf{X}_{\mathcal{S}} = \mathbf{x}_{\mathcal{S}})$ is the conditional probability distribution of $\mathbf{X}_{\overline{\mathcal{S}}}$ given that $\mathbf{X}_{\mathcal{S}} = \mathbf{x}_{\mathcal{S}}$. In this setting, the corresponding Shapley value, ϕ_k, for a feature k is given by

$$\phi_k(\mathbf{x}, \hat{y}) = \sum_{\mathcal{S} \subseteq \mathcal{M} \setminus \{k\}} \frac{|\mathcal{S}|!(M - |\mathcal{S}| - 1)!}{M!} \left[v_{\mathbf{x}, \hat{y}}(\mathcal{S} \cup \{k\}) - v_{\mathbf{x}, \hat{y}}(\mathcal{S}) \right], \tag{45}$$

with the value function defined as in (43). Notice that the Shapley values can be computed for *each* prediction of the model $\hat{y}$. We denote such explainable methods as *local*. With the definition of the value function in (43), the Shapley value for each feature in a given prediction can be interpreted as the *expected* change in the prediction model, over all subsets $\mathcal{S}$, when *including* feature k actively in the model compared to when feature k is marginalized in the prediction. The marginalization of feature k means that we regard the value of feature k as unknown, and so the corresponding conditional expectation of the prediction model will depend on the random variable X_k. By the construction of the Shapley values, we will also have

$$\sum_{k=1}^{M} \phi_k(\mathbf{x}, \hat{y}) = v(\mathcal{M}) - v(\phi) = \hat{y}(\mathbf{x}) - E_{\mathbf{X}}[\hat{y}(\mathbf{X})],$$

which shows that the sum of the Shapley values equals to the net change in the prediction compared to the expected prediction, and so the Shapley value for each feature can be interpreted as the portion of this net change that is due to this feature. Note that the Shapley value can be both positive and negative. This can also be interpreted as the *direction* in which feature k contributes to the prediction.

In a broader context, evaluating the properties of several explanation methods, in Lundberg and Lee (2017) the definition in (45) was defined as Shapley additive explanation values, abbreviated SHAP values, where additive explanation methods referred to the property that the sum of the explanation of each feature was equal to the difference between the prediction itself and the prediction $v(\phi) = \phi_0$. We will from hereon denote $\phi_k^{\text{SHAP}}(\mathbf{x}, \hat{y})$ as the corresponding Shapley value with value functions defined as in the SHAP framework.

8.1.1 Approximation methods

Even with the procedure of SHAP values, only needing to fit one model, the computations are still heavy as we need to iterate over all subsets, for each feature. In addition, the value functions defined in (43) are in general unknown, and need to be estimated. This also means that the SHAP values need to be estimated. A general procedure as outlined in Castro et al. (2009), and applied on regression models in Štrumbelj and Kononenko (2014) is to instead develop a resampling algorithm in which a value iteratively converges to the true Shapley estimate for a particular feature. Formally, use the definition of SHAP value as in (41) with

$$\phi_k^{\text{SHAP}}(\mathbf{x}, \hat{y}) = \frac{1}{M!} \sum_{R} \left[v_{\mathbf{x},\hat{y}}(s_k(R) \cup \{k\}) - v_{\mathbf{x},\hat{y}}(s_k(R)) \right]. \tag{46}$$

We will for the moment assume all features to be mutually independent, that is $E[X_j | X_k = x_k] = E[X_j]$ for all $j \neq k$. Given L data samples $\mathbf{x}_1, \ldots, \mathbf{x}_L$ for instance coming from the training data used to construct $\hat{y}$, then an estimate of (43), denoted $\hat{v}_{\mathbf{x},\hat{y}}(\mathcal{S})$, for a given subset $\mathcal{S}$ and observation $\mathbf{x}$ is given by

$$\hat{v}_{\mathbf{x},\hat{y}}(\mathcal{S}) = \frac{1}{L} \sum_{l=1}^{L} \hat{y}(\mathbf{x}_{\mathcal{S}}, \mathbf{x}^l_{\bar{\mathcal{S}}}), \tag{47}$$

where $\mathbf{x}_{\mathcal{S}}$ is the constant vector of features values from the subset of features $\mathcal{S}$ in the observation $\mathbf{x}$, while $\mathbf{x}^l_{\bar{\mathcal{S}}}$ denotes the observed feature values in the subset of features $\bar{\mathcal{S}}$ for sample l. The estimate $\hat{v}_{\mathbf{x},\hat{y}}(\mathcal{S})$ simply comes from the conditional sample mean estimator which is an unbiased estimator that converges in probability to the true conditional expectation as $L \to \infty$. Applying the estimate given in (47), consider the following iterative procedure: In each iteration, sample a particular reordering R_i of the features (with probability $1/M!$). Compute $\hat{v}^i_{\mathbf{x},\hat{y}}(s_k(R) \cup \{k\}) - \hat{v}^i_{\mathbf{x},\hat{y}}(s_k(R))$ based on L data samples for the training data. Perform I iterations, and do the same procedure. See Algorithm 1.

By the central limit theorem, the estimator of the approximation in Algorithm 1 asymptotically has a normal distribution with mean equal to the true Shapley estimate, and a variance which decreases inversely proportional with the number of iterations I (Štrumbelj & Kononenko, 2014). However, in order to compare the contribution between the features, one wants to estimate the SHAP value for each feature, and so using Algorithm 1 for each feature separately would be infeasible when the number of feature M is large. An alternative is to adaptively approximate all SHAP values in the same loop, using some criterion to reduce the total number of iterations needed to get satisfactory approximations. For instance in Štrumbelj and Kononenko (2014), a feature is randomly sampled, and the approximation of

Algorithm 1 Approximating SHAP value estimate for feature k

```
Given data samples $\mathbf{x}_1, \dots, \mathbf{x}_N$ (for instance training data), a model $\hat{y}(\mathbf{x})$, and a feature $k$.
$\hat{\phi}_k = 0$.
for $i = 1, 2, \dots, I$ do
    Sample a reordering $R_i$ of $\mathcal{M}$.
    Sample $L < N$ data samples.
    $\hat{\phi}_k = \hat{\phi}_k + \hat{v}^i_{\mathbf{x},\hat{y}}(s_k(R_i) \cup \{k\}) - \hat{v}^i_{\mathbf{x},\hat{y}}(s_k(R_i))$
end for
$\hat{\phi}_k = \frac{\hat{\phi}_k}{I}$
```

the SHAP value is updated by using Algorithm 1 with $I = 1$. However, when each feature has been updated sufficiently many times, the algorithm investigates the asymptotic variance of the estimator of the approximation for each feature, and chooses to update the SHAP value where the variance will reduce the most. See Algorithm 2 in Štrumbelj and Kononenko (2014).

Another approximation method is the *kernel SHAP* method introduced in Lundberg and Lee (2017), and based on the fact that computation of Shapley values can be seen as a minimization problem (Charnes et al., 1988):

$$\underset{(\phi_1,\dots,\phi_M)\in\mathcal{R}^M}{\arg\min} \sum_{\mathcal{S}\subseteq\mathcal{M}} \left(v(\mathcal{S}) - \left(\phi_0 + \sum_{j\in\mathcal{S}} \phi_j \right) \right)^2 k(M, \mathcal{S}), \tag{48}$$

where

$$k(M, \mathcal{S}) = \frac{M-1}{\binom{M}{|\mathcal{S}|}|\mathcal{S}|(M - |\mathcal{S}|)}$$

denote the *Shapley kernel weights*. As described in Aas et al. (2021), let Z denote the $2^M \times (M+1)$ matrix where the first column is one for each row, while the rest of the columns in any row is a binary representation (zero or one) of which features are included in a particular subset $\mathcal{S}$ out of all 2^M subsets. Further, we define the vector $\mathbf{v}$ of value functions and the $2^M \times 2^M$ diagonal matrix of Shapley kernel weights, with the order of the subsets $\mathcal{S}$ to be in the same order as the subsets $\mathcal{S}$ in each row of Z. Then the objective function in (48) can be written as

$$(\mathbf{v} - Z\boldsymbol{\phi})^T W (\mathbf{v} - Z\boldsymbol{\phi})$$

with solution

$$\boldsymbol{\phi} = (Z^T W Z)^{-1} Z^T W \mathbf{v} \tag{49}$$

Note that $k(M, M) = k(M, \emptyset) = \infty$, and so in practice one may set these as a large number instead. The Kernel SHAP aims to approximate the solution in (49), by construction of a resampling procedure, and then approximation of $\mathbf{v}$ by $\hat{\mathbf{v}}$ as given in (47). The resampling procedure is constructed by the fact that the value of $k(M, \mathcal{S})$ has a large variation for different subsets $\mathcal{S}$, and so several subsets contribute a small amount to the objective in (48). Therefore, in each iteration a subset $\mathcal{S}$ (not including $\emptyset$ and $\mathcal{M}$) is sampled (with replacement) with probability distribution according to the Shapley kernel weights. Given I such iterations, the kernel SHAP estimate is then given by

$$\hat{\boldsymbol{\phi}} = (Z_I^T W_I Z_I)^{-1} Z_I^T W_I \hat{\mathbf{v}}_I, \tag{50}$$

with Z_I, W_I and $\mathbf{v}_I$ are with respect to the I resampled subsets $\mathcal{S}$.

When the features are not mutually independent, the approximation given in (47) is no longer valid, and the complexity in the computations increases dramatically as one must in this case estimate conditional probability distributions. We refer to Aas et al. (2021) for several approaches on how to account for this situation.

8.1.2 SHAP values in tree ensemble models

The complexity in the SHAP values due to all the subsets $\mathcal{S}$ as well as the estimation of the value functions often makes the computations tiresome in large high-dimensional black-box models, with exponential running time. However, for one particular black-box model, namely tree ensemble models, Lundberg et al. (2020) showed that it was possible to compute the SHAP values in polynomial running time without having to do any resampling.

8.1.3 SHAP interaction values

The SHAP values can be generalized according to the Shapley interaction index from game theory (Fujimoto et al., 2006). Given a pair of features j and k, the interaction contribution, $\Phi_{j,k}^{\mathrm{SHAP}}(\mathbf{x}, \hat{y})$, from feature j and k other than their marginal contributions is given by

$$\Phi_{j,k}^{\mathrm{SHAP}}(\mathbf{x}, \hat{y}) = \sum_{\mathcal{S} \subseteq \mathcal{M} \backslash \{j,k\}} \frac{|\mathcal{S}|!(M - |\mathcal{S}| - 2)!}{2(M-1)!} \nabla_{j,k}(\mathcal{S}), \tag{51}$$

with

$$\nabla_{j,k}(\mathcal{S}) = v_{\mathbf{x},\hat{y}}(\mathcal{S} \cup \{j,k\}) - v_{\mathbf{x},\hat{y}}(\mathcal{S} \cup \{k\}) - [v_{\mathbf{x},\hat{y}}(\mathcal{S} \cup \{k\}) - v_{\mathbf{x},\hat{y}}(\mathcal{S})] \,.$$

We can interpret $\nabla_{j,k}(\mathcal{S})$ as the additional contribution in the prediction of including feature j actively and simultaneously together with feature k, compared to the contribution of feature k not including information of feature j. We define the *marginal* SHAP value of feature j, $\phi_{j,j}^{\mathrm{SHAP}}(\mathbf{x}, \hat{y})$, as

$$\phi_{j,j}^{\mathrm{SHAP}}(\mathbf{x}, \hat{y}) = \phi_{j}^{\mathrm{SHAP}}(\mathbf{x}, \hat{y}) - \sum_{k \neq j} \Phi_{j,k}^{\mathrm{SHAP}}(\mathbf{x}, \hat{y}),$$

and consequently we have that

$$\sum_{j=0}^{M} \sum_{k=0}^{M} \Phi_{j,k}^{\mathrm{SHAP}}(\mathbf{x}, \hat{y}) = \hat{y}(\mathbf{x}),$$

with $\Phi_{0,0}^{\mathrm{SHAP}}(\mathbf{x}, \hat{y}) = v_{\mathbf{x},\hat{y}}(\emptyset)$.

8.1.4 Explaining the model or explaining the data?

What is important to have in mind is that SHAP values are helpful to explain the predictions of the model. However, the model may not be reliable, and may not be a good representation of the underlying data generating process it is supposed to imitate. Covert et al. (2020) suggested an alternative to the SHAP value, referred to as SAGE value, in which the value function is defined not only as a function of the model, but also the underlying data through a loss function.

Part II

Motivation and summary of papers

Overview PhD

As part of the PhD, I collaborated together with the Gemini Center for Sepsis Research lead by Erik Solligård. The research team collaborated with Yale School of Public Health via Associate Professor in Epidemiology and Director at Yale Center for Perinatal, Pediatric and Environmental Epidemiology, Andrew Thomas DeWan. As a result I had a research stay at Yale School of Public Health in fall 2019. During the research stay, I got access to the computing infrastructure at Yale, specifically the Farnam cluster. The computing resources via Farnam was essential in order to be able to do the computations needed in all the papers on such high-dimensional data. Through the team of Andrew Thomas DeWan, I also got access to UK Biobank, a large long-term biobank consisting of around 500 000 participants each including genotype data. All three papers included the use of UK Biobank.

Paper 1

Saddlepoint approximations in binary genome-wide association studies

An essential problem in GWAS was the frequent observation that SNPs were declared significantly associated with a disease (which typically meant p-value less than 5×10^{-8}), however it could not be replicated in independent studies. It turned that a reason for this is that the p-values were simply not valid, as the normal approximation of the corresponding statistic was not sufficiently accurate. This was particularly observed for imbalanced binary phenotypes in Ma et al. (2013) and later Dey et al. (2017). It was also observed that SNPs with a small MAF, typically the most interesting SNPs, would also lead to a greater chance of invalid p-values, and therefore false positives.

In Dey et al. (2017), a saddlepoint approximation to the score test statistic was proposed, even being accurate far out in the tail of the distribution of the statistic. It was really surprising to us how accurate the saddlepoint approximation seemed to be based on their simulation. At first, we could not really understand the idea behind the transformation of the genotype vector, and why the maximum likelihood estimate under the null hypothesis was simply regarded as a plug-in constant. After reviewing the complex theory behind saddlepoint approximations, we discovered that the so-called *double saddlepoint approximation* would be a more intuitive procedure as we would then condition on the null hypothesis rather than assuming it to be true. After closer inspection, we also realized that Dey et al. (2017) assumed the probability distribution of the score test statistic to be continuous. However, when the allele count is simply integer-valued, we found out that the score test statistic actually is discrete, or specifically, has a *lattice* distribution. We therefore developed a continuity-corrected double saddlepoint approximation. Simultaneously, we managed to derive the exact probability distribution of the score test statistic for the model only including intercept as nuisance parameter, as well as the model with intercept and a binary covariate as nuisance parameters. Finally, we realized that the saddlepoint approximation proposed in Dey et al. (2017) actually was based on the *efficient score*, which really is a parameter transformation. By the theory of the efficient score, the corresponding score test statistic is asymptotically independent of the null hypothesis, which was the reason why the maximum likelihood estimate (at least asymptotically) could be treated as a constant when performing the saddlepoint approximation. Also here, we derived a continuity-corrected saddlepoint approximation when applying the efficient score.

By developing the double saddlepoint approximation for the score test statistic in a GWAS, as well as deriving the exact score test statistic in some scenarios, we could finally compare double saddlepoint approximation and the saddlepoint approximation using the efficient score proposed in Dey et al. (2017) with the exact distribution of the score test statistic. However, we found out that the evaluation of the different methods should be partitioned in what we refer to as conditionally valid tests and overall valid tests. By doing this we could conclude that continuity-correction is absolutely essential when the score test statistic has a lattice distribution. By simulations using a binary as well as a continuous covariate, we would also see that the double saddlepoint approximation appeared as having somewhat larger power than the efficient score in the case with imbalanced phenotypes as well as small MAFs.

Future work

Single variant tests are low-powered, particularly for variants with a small MAF, such as rare variants. However, such variants are of extra interest as many of them is situated in exons, regions inside the genes.

Several region-based tests have been proposed to increase the power. However, many of these region-based tests are based on several single-variant tests. At the best of our knowledge, there are still no region-based tests present taking into account the fact that the single-variant tests may include a statistic with a lattice distribution, and so continuity-corrections are needed. Even when using saddlepoint approximations, the region-based tests appear to be close to invalid or even invalid at the significance level Zhao et al. (2020).

Paper 2

A new method for exploring gene–gene and gene–environment interactions in GWAS with tree ensemble methods and SHAP values

One of the central topics in my PhD was so-called gene-gene and gene-environment interactions. GWAS has been successful in finding several single SNPs associated with several diseases, replicated in other studies (Visscher et al., 2017). However, one would still expect a larger genetic predisposition for a trait of disease than what was found via the estimated *heritability*, the observed variation in a trait or disease that could be explained by genetic variation, and not the environment. This is often called the missing heritability problem. One of the most frequent answers to why this is the case, is that the studies do not account for *interactions*, namely that the effects of two genes, or even between a gene and some environmental factor, is not simply additive with respect to the model output (Cox, 1984). However, incorporating such interactions in a generalized regression model in a genome-wide association study would exponentially increase the total number of tests to investigate. This will again lead to an even stricter rejection rule of the null hypothesis in order to control the false positive rate. Performing such genome-wide association studies has so far not been particularly successful. One might question whether the regression models are just to simple in order to model such complex relationships as interactions. With the rise of much more flexible machine learning models, the question was whether such models would be able to find these interactions. Tree ensemble models have been a popular alternative to the generalized regression models, such as random forests. In fact, the tree structures will automatically open up the possibilities to explore interactions. However, in this particular case, we expect only a small proportion of all SNPs to be relevant. The tree boosting models are known to be more suited for this setting. In addition, XGBoost, a tree boosting framework, had become very popular due to its smart and effective solutions both with respect to running time and memory capacity, making it a feasible software for high-dimensional data. However, even if the XGBoost model would give better predictions, they would still need to be interpreted. There have previously been several proposals on how to interpret such tree ensemble models (Lundberg et al., 2019), but with the introduction of SHAP values to explain machine learning models discussed in Section 8, Part I, several properties induced by the Shapley values turned out to be beneficial when fairly estimating the contribution of each feature in a model. The efficient TreeSHAP algorithm introduced in Lundberg et al. (2020) made it possible to explain XGBoost models based on high-dimensional data.

An important goal in Paper 2 was to investigate whether XGBoost models together with SHAP values could identify gene-gene and gene-environment interactions. However, before we could start with this, we would need to know whether this procedure would even find marginal effects from single SNPs such as in a regular GWAS. We therefore wanted to apply our procedure on a use-case where genome-wide association studies had been particularly successful in finding SNP associations, replicated in independent studies. Hence we chose to focus on obesity. Our procedure succeeded in finding the same SNP associations. We therefore moved on to develop a procedure for exploring interactions through Shapley interaction values. In this case, there were no known gene-gene interaction effects with respect to obesity. The results suggested that if there were any gene-gene interactions, they would be very small. What is important to have in mind is that our proposed procedure was intended to explore interaction effects. However, how reliable these suggested effects actually were in terms of *uncertainty*, was not known, and this became a natural next step that was investigated thoroughly in Paper 3.

Further work

Apart from developing methods to infer the uncertainty in the feature importance measures based on Shapley values, there are many other challenges specifically during the construction of the models. One challenge was the fact that all the features, SNPs and environmental features, needed to have small mutual correlations, or else the SHAP values would need to include estimated conditional probabilities in order to make reliable SHAP estimates (Aas et al., 2021). This will increase the computation time dramatically, and it would be interesting to consider efficient methods that can incorporate feature correlations in a high-dimensional setting when computing the SHAP values.

Paper 3

Inferring feature importance with uncertainties in high-dimensional data

A natural next step based on Paper 2 is to investigate how to infer the interaction effects including uncertainty of the features in XGBoost models using Shapley values. However, even inferring the uncertainty in marginal effects using Shapley values has not been investigated thoroughly yet so this became the focus.

What we realized in Paper 2, is that SHAP values are valuable when trying to explain which features the model considers important. However, the model can perform poorly, and it typically does for SNPs with small effects. Therefore, what the model considers important is not necessarily what actually should be considered important. We therefore went on to investigate alternatives to SHAP values that would not only bluntly investigate the model, but also take into account the data generating process the model was based on. The SAGE value introduced in Covert et al. (2020) differs from SHAP in the definition of the value function in the Shapley value context. In fact, the value function in SAGE is not only a function of the model, but also of the data it originates from via the loss function. We therefore consider this measure more reliable when the focus is to infer the actual feature importance from the unknown underlying data generation process. However, when going from SHAP to SAGE, the computation time needed increases dramatically, also for tree ensemble models, which makes it infeasible with high-dimensional data. We therefore introduced sub-SAGE, inspired by SAGE, but adjusted in order to be used for high-dimensional data.

Having a feature importance measure feasible to compute for high-dimensional data, the next step was how to infer the uncertainty in the sub-SAGE estimates. This meant we needed to find out actually what was stochastic and what was not when calculating the value functions. Due to the complexity in the sub-SAGE measure, we proposed a paired bootstrapping procedure in order to infer the uncertainty in the sub-SAGE estimates.

Even for sub-SAGE, efficient code is required. This was partly obtained through the R software, but also with help from the more efficient programming language C++, via the Rcpp package in R. Using the Rcpp package, the computation time of the part of the code including a recursive algorithm, dramatically reduced. By letting the many for-loops to be executed in C++ also improved the running time greatly.

Further work

The next natural step would be to move on to inferring feature importance for interactions. This could in principle be done by defining sub-SAGE interaction values in a similar fashion as SHAP interaction values with adjustments. However, this step would require even faster code, and therefore more of the R-code would have to be translated to C++ code.

References, Part I and II

Aas, K., Jullum, M., & Løland, A. (2021). Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. *Artificial Intelligence*, *298*.

Alsabti, K., Ranka, S., & Singh, V. (1998). Clouds: A decision tree classifier for large datasets. *KDD*, 2–8.

Barndorff-Nielsen, O. E. (1990). Approximate Interval Probabilities. *Journal of the Royal Statistical Society: Series B (Methodological)*, *52*(3), 485–496.

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32.

Butler, R. W. (2007). *Saddlepoint approximations with applications.* Cambridge University Press.

Bycroft, C., Freeman, C., Petkova, D., Band, G., Elliott, L. T., Sharp, K., Motyer, A., Vukcevic, D., Delaneau, O., O'Connell, J., Cortes, A., Welsh, S., Young, A., Effingham, M., McVean, G., Leslie, S., Allen, N., Donnelly, P., & Marchini, J. (2018). The UK Biobank resource with deep phenotyping and genomic data. *Nature*, *562*(7726), 203–209.

Casella, G., & Berger, R. (2001). *Statistical inference* (2nd ed.). Duxbury Resource Center.

Castro, J., Gómez, D., & Tejada, J. (2009). Polynomial calculation of the Shapley value based on sampling. *Computers & Operations Research*, *36*(5), 1726–1730.

Charnes, A., Golany, B., Keane, M., & Rousseau, J. (1988). In J. K. Sengupta & G. K. Kadekodi (Eds.), *Econometrics of Planning and Efficiency* (pp. 123–133). Springer Netherlands.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, 785–794.

Covert, I., Lundberg, S., & Lee, S.-I. (2020). Understanding global feature contributions with additive importance measures.

Cox, D. R. (1984). Interaction [Publisher: [Wiley, International Statistical Institute (ISI)]]. *International Statistical Review / Revue Internationale de Statistique*, *52*(1), 1–24.

Daniels, H. E. (1954). Saddlepoint Approximations in Statistics. *The Annals of Mathematical Statistics*, *25*(4), 631–650.

Daniels, H. E. (1987). Tail Probability Approximations. *International Statistical Review / Revue Internationale de Statistique*, *55*(1), 37–48.

Dey, R., Schmidt, E. M., Abecasis, G. R., & Lee, S. (2017). A fast and accurate algorithm to test for binary phenotypes and its application to PheWAS. *American Journal of Human Genetics*, *101*(1), 37–49.

Dudbridge, F., & Gusnanto, A. (2008). Estimation of significance thresholds for genomewide association scans. *Genetic Epidemiology*, *32*(3), 227–234.

Efron, B., & Tibshirani, R. J. (1994). *An Introduction to the Bootstrap*. Chapman & Hall/CRC.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, *38*(4), 367–378.

Fujimoto, K., Kojadinovic, I., & Marichal, J.-L. (2006). Axiomatic characterizations of probabilistic and cardinal-probabilistic interaction indices. *Games and Economic Behavior*, *55*(1), 72–99.

Goeman, J. J., & Solari, A. (2014). Multiple hypothesis testing in genomics. *Statistics in Medicine*, *33*(11), 1946–1978.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer New York.

Jin, R., & Agrawal, G. (2003). Communication and Memory Efficient Parallel Decision Tree Construction.

Li, P., Wu, Q., & Burges, C. J. (2008). McRank: Learning to Rank Using Multiple Classification and Gradient Boosting. In J. C. Platt, D. Koller, Y. Singer, & S. T. Roweis (Eds.), *Advances in Neural Information Processing Systems 20* (pp. 897–904). Curran Associates, Inc.

Lindsey, J. K. (1996). *Parametric statistical inference*. Oxford University Press.

Lipovetsky, S., & Conklin, M. (2001). Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, *17*(4), 319–330.

Loh, P.-R., Tucker, G., Bulik-Sullivan, B. K., Vilhjálmsson, B. J., Finucane, H. K., Salem, R. M., Chasman, D. I., Ridker, P. M., Neale, B. M., Berger, B., Patterson, N., & Price, A. L. (2015). Efficient Bayesian mixed model analysis increases association power in large cohorts. *Nature genetics*, *47*(3), 284–290.

Lugannani, R., & Rice, S. (1980). Saddle point approximation for the distribution of the sum of independent random variables. *Advances in Applied Probability*, *12*(2), 475–490. https://doi.org/10.2307/1426607

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, *2*(1).

Lundberg, S. M., Erion, G. G., & Lee, S.-I. (2019). Consistent Individualized Feature Attribution for Tree Ensembles. *arXiv:1802.03888 [cs, stat]*.

Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems 30* (pp. 4765–4774). Curran Associates, Inc.

Ma, C., Blackwell, T., Boehnke, M., & Scott, L. J. (2013). Recommended joint and meta-analysis strategies for case-control association testing of single low-count variants. *Genetic epidemiology*, *37*(6), 539–550.

Price, A. L., Patterson, N. J., Plenge, R. M., Weinblatt, M. E., Shadick, N. A., & Reich, D. (2006). Principal components analysis corrects for stratification in genome-wide association studies. *Nature Genetics*, *38*(8), 904–909.

Shapley, L. S. (1953). 17. A Value for n-Person Games. *Contributions to the Theory of Games (AM-28), Volume II* (pp. 307–318). Princeton University Press.

Skovgaard, I. M. (1987). Saddlepoint Expansions for Conditional Distributions. *Journal of Applied Probability*, *24*(4), 875–887.

Štrumbelj, E., & Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, *41*(3), 647–665.

Visscher, P. M., Wray, N. R., Zhang, Q., Sklar, P., McCarthy, M. I., Brown, M. A., & Yang, J. (2017). 10 Years of GWAS Discovery: Biology, Function, and Translation. *American Journal of Human Genetics*, *101*(1), 5–22.

Zhao, Z., Bi, W., Zhou, W., VandeHaar, P., Fritsche, L. G., & Lee, S. (2020). UK Biobank Whole-Exome Sequence Binary Phenome Analysis with Robust Region-Based Rare-Variant Test. *The American Journal of Human Genetics*, *106*(1), 3–12.

Zhou, W., Nielsen, J. B., Fritsche, L. G., Dey, R., Gabrielsen, M. E., Wolford, B. N., LeFaive, J., VandeHaar, P., Gagliano, S. A., Gifford, A., Bastarache, L. A., Wei, W.-Q., Denny, J. C., Lin, M., Hveem, K., Kang, H. M., Abecasis, G. R., Willer, C. J., & Lee, S. (2018). Efficiently controlling for case-control imbalance and sample relatedness in large-scale genetic association studies. *Nature Genetics*, *50*(9).

Part III

Research papers

Paper 1

Saddlepoint approximations in binary genome-wide association studies

Saddlepoint approximations in binary genome-wide association studies

Pål Vegard Johnsen[1,2], Øyvind Bakke[2], Thea Bjørnland[2], Andrew Thomas DeWan[3], and Mette Langaas[2]

[1]SINTEF Digital, Oslo, Norway
[2]Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway
[3]Department of Chronic Disease Epidemiology and Center for Perinatal, Pediatric and Environmental Epidemiology, Yale School of Public Health

Abstract

We investigate saddlepoint approximations applied to the score test statistic in genome-wide association studies with binary phenotypes. The inaccuracy in the normal approximation of the score test statistic increases with increasing sample imbalance and with decreasing minor allele count. Applying saddlepoint approximations to the score test statistic distribution greatly improve the accuracy, even far out in the tail of the distribution. By using exact results for an intercept model and binary covariate model, as well as simulations for models with nuisance parameters, we emphasize the need for continuity corrections in order to achieve valid p-values. The performance of the saddlepoint approximations is evaluated by overall and conditional type I error rate on simulated data. We investigate the methods further by using data from UK Biobank with skin and soft tissue infections as phenotype, using both common and rare variants. The analysis confirms that continuity correction is important particularly for rare variants, and that the normal approximation gives a highly inflated type I error rate for case imbalance.

1 Introduction

We consider score tests for logistic regression models in which the response is imbalanced and the covariate of interest is discrete and skewed. This typically occurs in a genome-wide association study (GWAS) with binary phenotypes, henceforth denoted binary GWAS, where one of the phenotypes is rare.

In a GWAS each single nucleotide polymorphism (SNP) is tested individually for association with a particular phenotype. In a modern biobank including several hundred thousands SNPs, rejection of the null hypothesis needs to be evaluated with a very low p-value threshold, typically equal to $5 \cdot 10^{-8}$, in order to control the family-wise error rate (FWER). In a binary GWAS with imbalanced response, new challenges arise.

As an example, we consider a follow-up study on skin and soft tissue infection (SSTI) using UK-biobank data, motivated by Rogne et al. (2021). Using data on unrelated white European individuals with no prior history of SSTI at recruitment, we obtain 6.5 years of follow-up data on approximately 300 000 individuals, out of which approximately 0.7% where diagnosed with SSTI during follow-up, and classified as cases. The overall sample size may be large, but if there are few cases or controls with a certain genotype, relying on asymptotic normality of the score test statistic may yield spurious results. In fact, the score test applied under asymptotic theory yields invalid p-values if the case proportion is too small. In addition, the severity in this flaw increases with decreasing minor allele frequencies (MAF). Both Ma et al. (2013) and Dey et al. (2017) have illustrated this issue for sample sizes of up to 20 000 individuals of which between 1% and 10% were cases. Motivated by the UK-Biobank SSTI data set, we show that the normal approximation can be flawed even when the total sample size is in the order of several hundred thousands. A solution proposed by Ma et al. (2013) is to apply the Firth (1993) bias-corrected logistic regression test. The test gives valid p-values when the imbalance is not too severe, and it is at the same time less conservative than the likelihood ratio test. As Firth's test is computationally inefficient for genome-wide testing, a test based on a saddlepoint approximation to the score statistic was proposed by Dey et al. (2017). This so-called SPA-test showed good properties yielding both valid or close to valid p-values even when Firth's test failed to do so, as well as being as powerful as Firth's test.

Our theoretical contribution to the ongoing development of valid score tests for genome-wide association studies with imbalanced binary phenotypes is twofold. First, we establish the discrete and bounded nature of the score, and derive the exact conditional distribution of the score test statistic for two particular examples of logistic regression models, namely models with intercept and genetic variant only, as well a models with an additional binary nuisance covariate (covariates associated with regression nuisance parameters). Second, we propose continuity-corrected saddlepoint approximations to the conditional distribution of the score statistic. We compare our proposed method against exact results as well as the approach introduced in Dey et al. (2017). We study the validity of tests both conditionally and unconditionally.

We show that a score test derived from the efficient score, or equivalently a null-orthogonal reparameterization of the logistic regression model, coincides with the SPA-test by Dey et al. (2017), thus providing a novel interpretation of the SPA-test as a two-step approximation to the conditional distribution of the score statistic.

We study our proposed continuity-corrected saddlepoint approximations as well as other existing methods, using the follow-up study of SSTIs as explained above, and on simulated data.

2 The score test statistic for logistic regression models in GWAS

2.1 Notation, statistical model and hypotheses

We consider tests for genotype–phenotype associations in large cohorts or populations. We assume that binary phenotypes, Y_i, non-genetic covariates $\boldsymbol{x}_i$ and allele counts g_i for a single variant, $i = 1, \ldots, n$, have been collected from n individuals. We consider directly biallelic allele counts in which $g_i \in \{0, 1, 2\}$. We model the relationship between the response and the covariates in a logistic regression model in which the Y_i are independent and Bernoulli distributed with success probability μ_i and

$$\operatorname{logit} \mu_i = \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{\beta} + \gamma g_i, \tag{1}$$

$i = 1, \ldots, n$. Here, $\boldsymbol{x}_i$ is a vector of dimension d containing 1 (corresponding to an intercept) and $d - 1$ covariates, $\boldsymbol{\beta}$ a d-dimensional vector of nuisance parameters and γ the parameter of interest. Our aim is to perform the hypothesis test

$$H_0 \colon \gamma = 0 \quad \text{against} \quad H_1 \colon \gamma \neq 0. \tag{2}$$

In a GWAS, the test is performed multiple times, for different genetic variants. To control the FWER at a 5% level in GWAS involving common variants, a significance level of $5 \cdot 10^{-8}$ is commonly used for each test (Jannot et al., 2015).

2.2 The score test statistic

The score vector is the gradient of the log-likelihood function with respect to the parameters, which for the logistic regression model (1) is

$$\boldsymbol{U} = \begin{pmatrix} \boldsymbol{U}_{\boldsymbol{\beta}} \\ U_\gamma \end{pmatrix} = \begin{pmatrix} X^{\mathrm{T}}(\boldsymbol{Y} - \boldsymbol{\mu}) \\ \boldsymbol{g}^{\mathrm{T}}(\boldsymbol{Y} - \boldsymbol{\mu}) \end{pmatrix}, \tag{3}$$

where $\boldsymbol{Y}$ and $\boldsymbol{g}$ are column vectors of length n with Y_i and g_i as elements respectively, $\boldsymbol{\mu} = E\boldsymbol{Y}$, and X is an $n \times d$ matrix with $\boldsymbol{x}_i^{\mathrm{T}}$ as rows. We have partitioned the score vector according to the parameter of interest, γ, and the nuisance parameters, $\boldsymbol{\beta}$. The score vector has mean $\mathbf{0}$ and covariance matrix, by definition referred to as the expected Fisher information

$$F = \begin{pmatrix} F_{\boldsymbol{\beta}\boldsymbol{\beta}} & \boldsymbol{F}_{\gamma\boldsymbol{\beta}}^{\mathrm{T}} \\ \boldsymbol{F}_{\gamma\boldsymbol{\beta}} & F_{\gamma\gamma} \end{pmatrix} = \begin{pmatrix} X^{\mathrm{T}}WX & X^{\mathrm{T}}W\boldsymbol{g} \\ \boldsymbol{g}^{\mathrm{T}}WX & \boldsymbol{g}^{\mathrm{T}}W\boldsymbol{g} \end{pmatrix}, \tag{4}$$

where W is a diagonal matrix with $\mu_i(1-\mu_i)$ as the *ii* entry.

Using the score test, the null hypothesis of (2) is rejected if there is sufficient distance between the null value $\gamma = 0$ and the maximum likelihood estimate of γ. To judge this distance, without actually calculating the estimate, one uses the partial derivative U_γ of the log-likelihood with respect to γ at $\gamma = 0$, along with the probability distribution of U_γ under the null. The proof of the following observation is given in Appendix A.

Observation 1. *When $g_i \in (0,1,2)$, the score U_γ with respect to γ is a bounded lattice random variable with support on $-\boldsymbol{g}^{\mathrm{T}}\boldsymbol{\mu}$, $1-\boldsymbol{g}^{\mathrm{T}}\boldsymbol{\mu}$, $2-\boldsymbol{g}^{\mathrm{T}}\boldsymbol{\mu}$, $\ldots$, $\boldsymbol{g}^{\mathrm{T}}\mathbf{1}-\boldsymbol{g}^{\mathrm{T}}\boldsymbol{\mu}$.*

Importantly, the score is – as in our situation – often a function of unknown nuisance parameters. Then, one may consider the *conditional* null distribution of the score for the parameter of interest, U_γ, given that the components of the score vector corresponding to the nuisance parameters are equal to zero, $\boldsymbol{U_\beta} = \mathbf{0}$ (see e.g. Smyth, 2003). In this conditional framework, the unknown nuisance parameters are equal to the corresponding maximum likelihood estimates calculated under the null hypothesis $\gamma = 0$, so that $U_\gamma = \boldsymbol{g}^{\mathrm{T}}(\boldsymbol{Y}-\hat{\boldsymbol{\mu}})$, where $\hat{\boldsymbol{\mu}}$ consists of the fitted values of the null model. However, this conditional score test statistic will still be a lattice random variable, yet with a narrower support than described in Observation 1. See Appendix B.

In many applications, one may approximate the distribution of the score vector $\boldsymbol{U}$ by a multivariate normal distribution with mean $\mathbf{0}$ and covariance matrix F. The conditional distribution of U_γ given $\boldsymbol{U_\beta} = \mathbf{0}$ under the null ($\gamma = 0$) is then asymptotically a normal distribution with mean 0 and variance

$$\tilde{F}_{\gamma\gamma} = \boldsymbol{g}^{\mathrm{T}}W\boldsymbol{g} - \boldsymbol{g}^{\mathrm{T}}WX(X^{\mathrm{T}}WX)^{-1}X^{\mathrm{T}}W\boldsymbol{g}. \tag{5}$$

As outlined in the Introduction, the normal approximation to the score vector may lead to spurious results for genotype–phenotype associations when the phenotype is a binary variable. For example, even if the the overall sample size is large, the normal approximation may be inaccurate if the sample contains few individuals with response $y_i = 1$ (e.g., having the disease under study) and genotype $g_i > 0$ (carrying the minor allele).

In the next section, we present a score test for (2) based on a double saddlepoint approximation to the conditional null distribution of the score statistic U_γ for the logistic regression model (1), given $\boldsymbol{U_\beta} = \mathbf{0}$. Here, we first state two observations that give the *exact* conditional null distribution for two special cases of the regression model (1). Proofs are given in Appendix A.

Observation 2. *Consider a logistic regression model as in* (1)*, but with* $\operatorname{logit}\mu_i = \beta + \gamma g_i$*, henceforth denoted the intercept model. Let n_j be the number of individuals with genotype $g_i = j$, $j = 0, 1, 2,$ and let* $\operatorname{logit}\mu = \beta$*. Then, the null distribution of U_γ given $U_\beta = 0$ is a sum of trivariate hypergeometric point probabilities,*

$$P(U_\gamma = u \mid U_\beta = 0) = \sum_{(v_0,v_1,v_2)\in S} \frac{\binom{n_0}{v_0}\binom{n_1}{v_1}\binom{n_2}{v_2}}{\binom{n}{n\mu}} = \sum_{k=\max(\lceil (u^*-n_1)/2\rceil,0)}^{\min(\lfloor u^*/2\rfloor,n_2)} \frac{\binom{n_0}{n\mu-u^*+k}\binom{n_1}{u^*-2k}\binom{n_2}{k}}{\binom{n}{n\mu}},$$

where the sum is taken over all triples (v_0, v_1, v_2) of integers in the set S defined by $0 \le v_j \le n_j$ for $j = 0,\ 1,\ 2,\ \ v_0 + v_1 + v_2 = n\mu$ and $v_1 + 2v_2 = u^*$, and $u^* = u + (n_1 + 2n_2)\mu$. The function outputs $\lceil x\rceil$ and $\lfloor x\rfloor$ denote the least integer greater than or equal to x (ceiling), and the largest integer less than or equal to x (floor) respectively.

Observation 3. *Consider a logistic regression model as in* (1)*, where* $\operatorname{logit}\mu_i = \beta_0 + \beta_1 x_i + \gamma g_i$*, and x_i is a binary covariate taking value* 0 *or* 1 (*model with intercept and one binary non-genetic covariate*)*. Let l_j be the number of individuals with $x_i = 0$ and genotype $g_i = j$, $j = 0,\ 1,\ 2,$ and let $l = l_0 + l_1 + l_2$. Define similar counts m_j and m for individuals with $x_i = 1$. Let* $\operatorname{logit}\mu_0 = \beta_0$*, and* $\operatorname{logit}\mu_1 = \beta_0 + \beta_1$*. Then, under the null hypothesis,*

$$P(U_\gamma = u \mid \boldsymbol{U_\beta} = \mathbf{0}) = \sum_{\boldsymbol{s}\in S} \frac{\binom{l_0}{v_0}\binom{l_1}{v_1}\binom{l_2}{v_2}}{\binom{l}{l\mu_0}} \frac{\binom{m_0}{w_0}\binom{m_1}{w_1}\binom{m_2}{w_2}}{\binom{m}{m\mu_1}},$$

where the sum is taken over all sextuples $\boldsymbol{s} = (v_0, v_1, v_2, w_0, w_1, w_2)$ of integers in the set S defined by $0 \le v_j \le l_j,\ \ 0 \le w_j \le m_j$ for $j = 0,\ 1,\ 2,\ \ v_0 + v_1 + v_1 = l\mu_0,\ \ w_0 + w_1 + w_2 = m\mu_1$ and $v_1 + 2v_2 - (l_1 + 2l_2)\mu_0 + w_1 + 2w_2 - (m_1 + 2m_2)\mu_1 = u$.

From Observations 2 and 3, it follows that an *exact* p-value for the hypothesis test (2) can be computed for these two special cases of the logistic regression model (1). An extension of Observation 3 can also be derived for regression models with more categorical covariates. However, for more complex covariate patterns, this approach becomes computationally infeasible, or even intractable when continuous covariates are included. The next section introduces a method of computing p-values using double saddlepoint approximation.

3 Double saddlepoint approximation

Tail probabilities $P(U_\gamma \geq u \mid \boldsymbol{U}_{\boldsymbol{\beta}} = \boldsymbol{0})$ may be estimated by *double saddlepoint approximation* (Butler, 2007). This will require the *cumulant generating function* of $\boldsymbol{U} = \big(\boldsymbol{U}_{\boldsymbol{\beta}}^{\mathrm{T}} \quad U_\gamma\big)^{\mathrm{T}} = \big(X \quad \boldsymbol{g}\big)^{\mathrm{T}}(\boldsymbol{Y} - \boldsymbol{\mu})$ (Section 2.2) and of $\boldsymbol{U}_{\boldsymbol{\beta}}$.

3.1 Cumulant generating function

The joint cumulant generating function of $\boldsymbol{U}$ is defined by $K(\boldsymbol{t}) = \ln E\left(e^{\boldsymbol{t}^{\mathrm{T}}\boldsymbol{U}}\right)$, were $\boldsymbol{t}$ is a vector of dimension $d+1$. By using the fact that Y_i is Bernoulli distributed with parameter μ_i (Section 2.1), we obtain

$$K(\boldsymbol{t}) = \sum_{i=1}^{n}\Big(\ln\big(1-\mu_i+\mu_i e^{\boldsymbol{t}^{\mathrm{T}}\boldsymbol{z}_i}\big) - \mu_i \boldsymbol{t}^{\mathrm{T}}\boldsymbol{z}_i\Big), \tag{6}$$

$$\nabla K(\boldsymbol{t}) = \sum_{i=1}^{n}\mu_i\left(\frac{1}{(1-\mu_i)e^{-\boldsymbol{t}^{\mathrm{T}}\boldsymbol{z}_i}+\mu_i} - 1\right)\boldsymbol{z}_i, \quad \text{and} \tag{7}$$

$$H(\boldsymbol{t}) = \sum_{i=1}^{n}\frac{\mu_i(1-\mu_i)e^{-\boldsymbol{t}^{\mathrm{T}}\boldsymbol{z}_i}}{\big((1-\mu_i)e^{-\boldsymbol{t}^{\mathrm{T}}\boldsymbol{z}_i}+\mu_i\big)^2}\boldsymbol{z}_i\boldsymbol{z}_i^{\mathrm{T}}, \tag{8}$$

where ∇K and H denote the gradient and the Hessian of K, respectively, and $\boldsymbol{z}_i = \big(\boldsymbol{x}_i^{\mathrm{T}} \quad g_i\big)^{\mathrm{T}}$. The cumulant generating function of $\boldsymbol{U}_{\boldsymbol{\beta}}$, its gradient and Hessian, $K_{\boldsymbol{\beta}}$, $\nabla K_{\boldsymbol{\beta}}$ and $H_{\boldsymbol{\beta}}$, respectively, are obtained by replacing $\boldsymbol{z}_i$ by $\boldsymbol{x}_i$ and letting $\boldsymbol{t}$ have dimension d in (6)–(8).

3.2 Approximated tail probabilities with continuity correction

The survival function (right-tail probability) $S(u) = P(U_\gamma \geq u \mid \boldsymbol{U}_{\boldsymbol{\beta}} = \boldsymbol{0})$ can be approximated as given by Barndorff-Nielsen (1990),

$$\hat{S}(u) = 1 - \Phi\Big(w - \frac{1}{w}\ln\frac{v}{w}\Big), \tag{9}$$

where Φ denotes the standard normal cumulative distribution function. To approximate the conditional survival function of a lattice random variable we have chosen the double saddlepoint survival approximation with the so-called second continuity correction. Using $f(\boldsymbol{t}_1, \boldsymbol{t}_2)$ as shorthand for $f\big(\big(\boldsymbol{t}_1^{\mathrm{T}} \quad \boldsymbol{t}_2^{\mathrm{T}}\big)^{\mathrm{T}}\big)$, where f is a function and $\boldsymbol{t}_1$, $\boldsymbol{t}_2$ vectors, we have

$$w = \operatorname{sgn}(\hat{t}_\gamma)\sqrt{2\Big(-K(\hat{\boldsymbol{t}}_{\boldsymbol{\beta}}, \hat{t}_\gamma) + \hat{t}_\gamma\Big(u - \frac{1}{2}\Big)\Big)} \quad \text{and}$$

$$v = 2\Big(\sinh\frac{\hat{t}_\gamma}{2}\Big)\sqrt{\frac{\det H(\hat{\boldsymbol{t}}_{\boldsymbol{\beta}}, \hat{t}_\gamma)}{\det H_{\boldsymbol{\beta}}(\boldsymbol{0})}},$$

where $\big(\hat{\boldsymbol{t}}_{\boldsymbol{\beta}}^{\mathrm{T}} \quad \hat{t}_\gamma\big)^{\mathrm{T}}$ is the *saddlepoint* satisfying $\nabla K(\hat{\boldsymbol{t}}_{\boldsymbol{\beta}}, \hat{t}_\gamma) = \big(\boldsymbol{0}^{\mathrm{T}} \quad u - 1/2\big)^{\mathrm{T}}$ (Skovgaard, 1987, see Butler, 2007, p.114). In general, also the d-dimensional vector $\tilde{\boldsymbol{t}}_{\boldsymbol{\beta}}$ satisfying $\nabla K_{\boldsymbol{\beta}}(\tilde{\boldsymbol{t}}_{\boldsymbol{\beta}}) = \boldsymbol{0}$ is involved in the expressions for w and v, but $\tilde{\boldsymbol{t}}_{\boldsymbol{\beta}} = \boldsymbol{0}$ in our case (see Appendix C). Left-tail probabilities can be approximated, taking into account that U_γ is a lattice variable with step 1, by $P(U_\gamma \leq u \mid \boldsymbol{U}_{\boldsymbol{\beta}} = \boldsymbol{0}) = 1 - S(u+1)$.

3.3 Two-sided p-values

By assuming the score test statistic to have a normal distribution, and for some observation u, a two-sided p-value is reasonable and given by $P(|U_\gamma| \geq |u| \mid \boldsymbol{U_\beta} = \mathbf{0})$ (under the null). However, as the score test statistic has a lattice distribution, the point $-u$ might not be on the grid. If so, the closest grid point to $-u$ farthest away from zero is obtained by $u_{inv} = u - \text{sgn}(u) \cdot \lceil 2 \cdot |u| \rceil$. We define a two-sided p-value, assuming u positive, to be $P(U_\gamma \geq u \mid \boldsymbol{U_\beta} = \mathbf{0}) + P(U_\gamma \leq u_{inv} \mid \boldsymbol{U_\beta} = \mathbf{0})$, and vice versa when u is negative.

An example is given in Figure 1a where the exact lattice distribution of the score test statistic under the null hypothesis is given for the intercept model with a genotype vector simulated with MAF $= 0.05$ and a case proportion of 0.05 ($n = 1000$). Included is the support of the lattice distribution $[u^{min}, u^{max}] = [-5.5, 46.5]$. An observed $u = 4.5$ will then give $u_{inv} = -4.5$, a situation where $u_{inv} = -u$. The p-value is then equal to the sum of the bars coloured in orange. The deviation from the normal distribution increases for decreasing case proportion, as can be seen when comparing Figure 1a to 1b, where the case proportion is reduced to 0.01 while keeping the same genotype vector in Figure 1b. In fact, the skewness increases for decreasing case proportion such that the probability mass of the distribution is concentrated on the left, with a longer right tail. Consequently, the score test statistic is asymmetric as well as bounded, which means the point u_{inv} might be outside the support of the lattice distribution. In that case, a one-sided p-value will be computed as seen in Figure 1b with bars coloured orange only to the right of the observed $u = 1.9$ ($u_{inv} = -2.1 < u^{min} = -1.1$). The same observation of increased skewness can be seen with a fixed case proportion, but decreasing MAF.

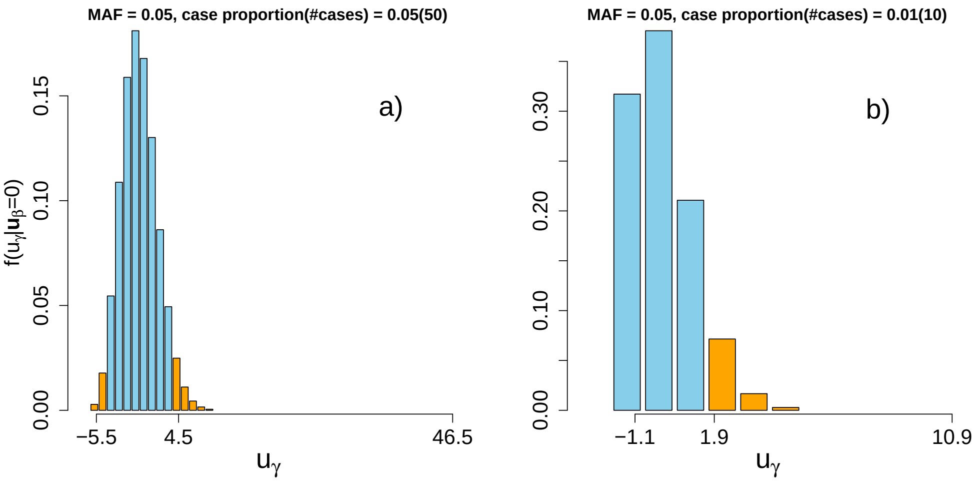


Figure 1: The exact lattice distribution of the score test statistic for the intercept model for different case proportions (genotype vector fixed, 1000 individuals). Included is the support $[u^{min}, u^{max}]$ of the lattice distribution in each case together with an example of an observed statistic in between, as well as the corresponding computed p-value coloured in orange. The deviation from normal distribution increases for decreasing case proportion. When the distribution is sufficiently skewed, a one-sided p-value is computed.

4 Single saddlepoint approximation using the efficient score

Our proposed method is related to the SPA-test by Dey et al. (2017), which is also based on a saddlepoint approximation to the distribution of a score test statistic. In this section, we provide a novel interpretation of the SPA-test as a two-step approximation to conditional inference, and propose a modification.

We implicitly introduced the score test statistic $\boldsymbol{g}^T(\boldsymbol{Y} - \hat{\boldsymbol{\mu}})$, where $\hat{\boldsymbol{\mu}}$ is the maximum likelihood estimate of $\boldsymbol{\mu}$ under the null hypothesis, solved by $\boldsymbol{U_\beta} = \mathbf{0}$. Rather than approximating the distribution of this test statistic directly, the common procedure for score test statistics in the presence of nuisance parameters is to use conditional inference by conditioning U_γ on $\boldsymbol{U}_{\boldsymbol{\beta}} = \mathbf{0}$.

Other methods for approximate conditional inference in the presence of nuisance parameters include orthogonal parametrization (Cox & Reid, 1987) and projective methods (Waterman & Lindsay, 1996). The first-order projective score, perhaps better known as the *efficient score*, is for our model (Equation (1)) defined by

$$\tilde{U}_\gamma = U_\gamma - \boldsymbol{F}_{\gamma\boldsymbol{\beta}} F_{\boldsymbol{\beta\beta}}^{-1} \boldsymbol{U}_{\boldsymbol{\beta}}.$$

As noted by Bickel et al. (1993), the efficient score may be interpreted in general as the score corresponding to a reparameterization $(\boldsymbol{\beta}, \gamma) \to (\boldsymbol{\alpha}, \gamma)$, by letting $\boldsymbol{\beta}(\boldsymbol{\alpha}, \gamma) = \boldsymbol{\alpha} - F_{\boldsymbol{\beta\beta}}^{-1} \boldsymbol{F}_{\gamma\boldsymbol{\beta}}^{T} \gamma$. With this reparameterization of the logistic regression model, $\text{logit}(\mu_i) = \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{\beta}(\boldsymbol{\alpha}, \gamma) + \gamma g_i = \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{\alpha} + \gamma \tilde{g}_i$, where $\tilde{g}_i = g_i - \boldsymbol{x}_i^{\mathrm{T}} F_{\boldsymbol{\beta\beta}}^{-1} \boldsymbol{F}_{\gamma\boldsymbol{\beta}}^{\mathrm{T}}$. Let $\tilde{F}$ denote the expected Fisher information of $\tilde{\boldsymbol{U}} = \left(\tilde{\boldsymbol{U}}_{\boldsymbol{\alpha}}^{\mathrm{T}} \quad \tilde{U}_\gamma\right)^{\mathrm{T}}$, the reparameterized score vector. With this reparameterization, the parameter γ and the nuisance parameters $\boldsymbol{\alpha}$ are locally information orthogonal at $\gamma = 0$, which means that $\tilde{\boldsymbol{F}}_{\boldsymbol{\alpha}\gamma}$ and $\tilde{\boldsymbol{F}}_{\gamma\boldsymbol{\alpha}}$ in the expected Fisher information $\tilde{F}$ are zero-vectors (see e.g. Lindsey (1996)). In this case, asymptotically $\tilde{\boldsymbol{U}}$ has a normal distribution, however additionally $\text{Cov}(\tilde{\boldsymbol{U}}_{\boldsymbol{\alpha}}(\hat{\boldsymbol{\mu}}), \tilde{U}_\gamma(\hat{\boldsymbol{\mu}})) \to \mathbf{0}$ when $\gamma = 0$ and $\boldsymbol{\mu} = \hat{\boldsymbol{\mu}}$. With $\tilde{\boldsymbol{U}}$ asymptotically multivariate normal, so will $\tilde{\boldsymbol{U}}_{\boldsymbol{\alpha}}$ and $\tilde{U}_\gamma$ (univariate) be. As covariance equal to zero for two normal distributed random variables implies independence, this means that the statistic of $\tilde{U}_\gamma$ conditional on $\tilde{\boldsymbol{U}}_{\boldsymbol{\alpha}} = \mathbf{0}$ is asymptotically the same as the unconditional distribution of $\tilde{U}_\gamma$ when the null hypothesis is true with $\hat{\boldsymbol{\mu}}$ treated as a plug-in constant for $\boldsymbol{\mu}$.

In our case with expected Fisher information given in (4),

$$\begin{aligned}
\tilde{U}_\gamma &= \boldsymbol{g}^{\mathrm{T}}(\boldsymbol{Y} - \boldsymbol{\mu}) - \boldsymbol{g}^{\mathrm{T}} W X (X^{\mathrm{T}} W X)^{-1} X^{\mathrm{T}} (\boldsymbol{Y} - \boldsymbol{\mu}) \\
&= (\boldsymbol{g}^{\mathrm{T}} - \boldsymbol{g}^{\mathrm{T}} W X (X^{\mathrm{T}} W X)^{-1} X^{\mathrm{T}})(\boldsymbol{Y} - \boldsymbol{\mu}) \\
&= (\boldsymbol{g} - X (X^{\mathrm{T}} W X)^{-1} X^{\mathrm{T}} W \boldsymbol{g})^{\mathrm{T}} (\boldsymbol{Y} - \boldsymbol{\mu}) \\
&= \tilde{\boldsymbol{g}}^{\mathrm{T}} (\boldsymbol{Y} - \boldsymbol{\mu}),
\end{aligned}$$

with $\tilde{\boldsymbol{g}} = \boldsymbol{g} - X(X^T W X)^{-1} X^T W \boldsymbol{g}$ the vector of all components $\tilde{g}_i$, and defined as in Dey et al. (2017). Observe that when $\boldsymbol{U}_{\boldsymbol{\beta}} = X^{\mathrm{T}}(\boldsymbol{Y} - \boldsymbol{\mu}) = \mathbf{0}$, the observed efficient score, $\tilde{u}$, is equal to u, the original observed score. Moreover, $E(\tilde{U}_\gamma | \tilde{\boldsymbol{U}}_{\boldsymbol{\alpha}} = \mathbf{0}) = E(U_\gamma | \boldsymbol{U}_{\boldsymbol{\beta}} = \mathbf{0}) = 0$, and $\text{Var}(\tilde{U}_\gamma | \tilde{\boldsymbol{U}}_{\boldsymbol{\alpha}} = \mathbf{0}) = \text{Var}(U_\gamma | \boldsymbol{U}_{\boldsymbol{\beta}} = \mathbf{0}) = \tilde{\boldsymbol{g}}^{\mathrm{T}} W \tilde{\boldsymbol{g}}$ with $\tilde{\boldsymbol{U}}_{\boldsymbol{\alpha}} = \boldsymbol{U}_{\boldsymbol{\beta}}$ under the null hypothesis. At last, observe that asymptotically as $\hat{\boldsymbol{\mu}} \xrightarrow{p} \boldsymbol{\mu}$ under the null hypothesis,

$$\begin{aligned}
&\text{Cov}(\tilde{U}_\gamma(\hat{\boldsymbol{\mu}}), \tilde{\boldsymbol{U}}_{\boldsymbol{\alpha}}(\hat{\boldsymbol{\mu}})) = F_{\gamma\boldsymbol{\alpha}}(\hat{\boldsymbol{\mu}}) = E\left(\tilde{U}_\gamma(\hat{\boldsymbol{\mu}}) \tilde{\boldsymbol{U}}_{\boldsymbol{\alpha}}(\hat{\boldsymbol{\mu}})^{\mathrm{T}}\right) \\
&= E\left(\left(\boldsymbol{g} - X\left(X^{\mathrm{T}} \hat{W} X\right)^{-1} X^{\mathrm{T}} \hat{W} \boldsymbol{g}\right)^{\mathrm{T}} (\boldsymbol{Y} - \hat{\boldsymbol{\mu}}) (\boldsymbol{Y} - \hat{\boldsymbol{\mu}})^{\mathrm{T}} X\right) \\
&= \boldsymbol{g}^{\mathrm{T}} E\left((\boldsymbol{Y} - \hat{\boldsymbol{\mu}}) (\boldsymbol{Y} - \hat{\boldsymbol{\mu}})^{\mathrm{T}}\right) X - E\left(\boldsymbol{g}^{\mathrm{T}} \hat{W} X \left(X^{\mathrm{T}} \hat{W} X\right)^{-1} X^{\mathrm{T}} (\boldsymbol{Y} - \hat{\boldsymbol{\mu}}) (\boldsymbol{Y} - \hat{\boldsymbol{\mu}})^{\mathrm{T}}\right) X \\
&\to \boldsymbol{g}^{\mathrm{T}} W X - \boldsymbol{g}^{\mathrm{T}} W X \left(X^{\mathrm{T}} W X\right)^{-1} X^{\mathrm{T}} W X = \mathbf{0}^T,
\end{aligned}$$

where $\hat{W}$ is the diagonal matrix with $\hat{\mu}_i(1 - \hat{\mu}_i)$ as the *ii* entry. Hence, we have shown indeed that $\tilde{U}_\gamma$ and $\tilde{\boldsymbol{U}}_{\boldsymbol{\alpha}}$ are asymptotically independent under the null hypothesis.

Under the null hypothesis, using $\tilde{\boldsymbol{U}}$ leads asymptotically to the same *unconditional* inference of $\tilde{U}_\gamma(\hat{\boldsymbol{\mu}})$ as the *conditional* inference of U_γ given $\boldsymbol{U}_{\boldsymbol{\beta}} = \mathbf{0}$. In other words, $f(\tilde{U}_\gamma) \xrightarrow{d} N(0, \tilde{F}_{\gamma\gamma})$, with $\tilde{F}_{\gamma\gamma}$ given in (5). However, this will still be inaccurate for an imbalanced response and a skewed covariate of interest. Under this framework, we interpret the test proposed by Dey et al. (2017) as a two-step approach, where the first step is to apply the efficient score, and in the second step the corresponding unconditional statistic is approximated by a single saddlepoint method via the univariate cumulant generating function of $\tilde{U}_\gamma$, given by

$$K(t) = \sum_{i=1}^{n} \ln(1 - \hat{\mu}_i + \hat{\mu}_i e^{\tilde{g}_i t}) - t\, \tilde{\boldsymbol{g}}^T \hat{\boldsymbol{\mu}}.$$

Since such a two-step approach does not require a double saddlepoint approximation, this method is computationally more efficient. In Dey et al. (2017), the efficient score test statistic is assumed to have a continuous distribution. However, when $g_i \in \{0, 1, 2\}$, the efficient score test statistic in fact has a lattice distribution. Therefore, we propose to use a continuity correction. Similarly to the continuity-corrected double saddlepoint method outlined in the previous section, left-tail probabilities are estimated as in Equation (9), now with

$$w = \mathrm{sgn}(\hat{t})\sqrt{2(\hat{t}(u_\gamma - 1/2) - K(\hat{t}))}, \text{ and } v = 2\mathrm{sinh}(\hat{t}/2)\sqrt{K''(\hat{t})},$$

where $\hat{t}$ is the saddlepoint obtained by solving $K'(\hat{t}) = u_\gamma - 1/2$. Furthermore, we apply the same algorithm for obtaining two-sided p-values as in Section 3.3.

5 Comparison of methods

For a specified significance level α, a *valid* test satisfies $P(\text{type I error}) \leq \alpha$. In our setting, we find it relevant to distinguish between conditional and overall (unconditional) validity. To clarify what is meant by this, consider a simple logistic regression model with no nuisance covariates (intercept only model). The covariate vector $\boldsymbol{g}$ is fixed while the response vector $\boldsymbol{Y}$ is random. Under the null, $Y_i \sim \mathrm{binom}(1, \mu)$ for all $i = 1, \dots, n$, where $\mu = \exp(\beta_0)/(1 + \exp(\beta_0))$. For a particular realization $\boldsymbol{y}$, the observed score test statistic $u_\gamma = \boldsymbol{g}^T(\boldsymbol{y} - \hat{\boldsymbol{\mu}}) = \boldsymbol{g}^T(\boldsymbol{y} - \bar{y}\mathbf{1})$ may be compared to the conditional null distribution of U_γ, i.e. the distribution of $\boldsymbol{g}^T(\boldsymbol{Y} - \bar{y}\mathbf{1})$ given that $\boldsymbol{Y}$ is restricted by $\sum_i Y_i = n\bar{y} = v$ (Observation 2). Thus, for all datasets in which the realization $\boldsymbol{y}$ satisfies $\sum_i y_i = v$, a test is *conditionally* valid only when $P(\text{type I error}|\sum_i Y_i = v) \leq \alpha$. On the other hand, the *overall* probability of type I error is given by

$$\sum_v \left[P\left(\text{type I error} \mid \sum_{i=1}^n Y_i = v\right) P\left(\sum_{i=1}^n Y_i = v\right)\right]. \tag{10}$$

A test that is conditionally valid for all v, will also be valid overall. The exact test derived in Observation 2 satisfies this property. An approximation to the exact test may be conditionally valid for some v, but invalid overall, or valid overall but conditionally invalid for some v. In the case with nuisance covariates, Equation (10) may be generalized to:

$$\sum_{X, \boldsymbol{y} \,:\, \boldsymbol{U_\beta} = \mathbf{0}} \left[P\left(\text{type I error} \mid \boldsymbol{U_\beta} = \mathbf{0}\right) P\left(\boldsymbol{U_\beta} = \mathbf{0}\right)\right].$$

To evaluate the performance of our proposed methods, we consider both conditional and overall validity for models where the exact test is available. Approximation methods are evaluated based on their ability to control the overall type I error rate as well as the proportion of tests that are conditionally invalid.

5.1 Intercept model

In this section, we consider the intercept model with no nuisance parameters. We compare two discrete and two continuous conditional inference approximation methods with the exact test. The discrete methods are the double saddlepoint method with continuity correction as described in section 3, henceforth termed DSPA-CC, and the continuity-corrected single saddlepoint method based on the efficient score as described in section 4, henceforth termed ESPA-CC. The continuous methods are the normal approximation and the single saddlepoint method based on the efficient score (henceforth termed ESPA). To the best of our knowledge, the ESPA method mimics the SPA-test of Dey et al. (2017) as implemented in the SPA-package in R. We present a simple example in order to highlight some of the key differences between the methods.

Let $n = 1000$ and let $\boldsymbol{g}$ be the covariate vector with $n_0 = 980$ and $n_1 = 20$ and $n_2 = 0$. Without specifying what μ is, we first calculate the probabilities $P\left(\text{type I error} \mid \sum_{i=1}^n Y_i = v\right)$, for all $v = 1, 2, \dots, n-1$. For a particular realization v, and discrete sample space within the support $[u_L, u_U]$ of the conditional null distribution of U_γ, where u_L and u_U need not be integers, we obtain the rejection region

$\{u_L, \ldots, c_L\} \cup \{c_U, \ldots, u_U\}$ of the exact test. This can be achieved by a grid search from the left to obtain c_L as well as a separate grid search from the right to obtain c_U since the probability distribution is not symmetric. Then, $P\left(\text{type I error} \mid \sum_{i=1}^n Y_i = v\right) = P(U_\gamma \leq c_L \cup U_\gamma \geq c_U \mid \sum_{i=1}^n Y_i = v)$. For the approximation methods DSPA-CC, SPA-CC and SPA, we similarly use a grid search to identify lower (c_L^*) and upper (c_U^*) critical values that lead to rejection at the specified significance level. For the normal approximation, we obtain a critical value c^* from the normal distribution with mean 0 and variance $\frac{v}{n}(1-\frac{v}{n})\left[n_0(0-\frac{n_1}{n})^2 + n_1(1-\frac{n_1}{n})^2\right]$, and then obtain the proper lower and upper critical values by the nearest grid points c_L^* and c_U^* to $-c^*$ and c^* such that $c_L^* \leq -c^*$ and $c_U^* \geq c^*$. Then, for rejection regions $\{u_L, \ldots, c_L^*\} \cup \{c_U^*, \ldots, u_U\}$, we calculate the exact conditional probability of erroneously rejecting the null hypothesis using the different approximation methods. For a specified value of μ, we obtain probabilities $P\left(\sum_{i=1}^n Y_i = v\right)$ for each observed v. The overall probability of type I error can be computed according to Equation (10). In addition, the probability of a conditionally invalid test for each method and for each μ can be computed by observing which values v where $P(\text{type I error} \mid \sum_i Y_i = v) > \alpha$, and add together the probabilities $P\left(\sum_{i=1}^n Y_i = v\right)$ for each such v. See Figure 2.

From this example, we make four observations;

1. The exact test is always conservative (see Figure 2). When a significance level α is specified, the discrete nature of the test results in an achieved significance level less than α. This observation is of course well-known for discrete test statistics.

2. Both of the discrete approximations (DSPA-CC and SPA-CC) closely resemble the exact test in terms of overall type I error rates (Figure 2). At significance level $\alpha = 0.05$, both methods gave conditionally invalid tests in four situations; $v = 301$, $v = 325$, $v = 675$, and $v = 699$. For instance for $\mu = 0.31$ and $\mu = 0.69$, this results in probabilities ≈ 0.04 of sampling a dataset where these methods are conditionally invalid. At significance level $\alpha = 5 \cdot 10^{-5}$, DSPA-CC is conditionally valid for any v, while SPA-CC is conditionally invalid for $v = 406$ and $v = 594$. For instance for $\mu = 0.41$ and $\mu = 0.59$, this results in a slight probability (≈ 0.02) of sampling a dataset where the SPA-CC method is conditionally invalid.

3. Even at significance level $\alpha = 0.05$, the normal approximation is invalid for different μ-values (Figure 2). For significance level $5 \cdot 10^{-5}$, the normal approximation is valid when the response is balanced ($\mu \approx 0.5$). However, for skewed responses (small or large μ), the normal approximation becomes severely unreliable. At significance level $\alpha = 0.05$, the normal approximation was conditionally invalid in around 40% of possible realizations of $\sum_i Y_i$. At significance level $\alpha = 5 \cdot 10^{-5}$, this number had increased to around 64%. The majority of situations where the normal approximation was conditionally invalid was for small or large number of cases v, which is in-line with the observations made of overall type I error rates for skewed responses (Figure 2).

4. The SPA method is less conservative than the exact test, and at times anti-conservative. At significance level $\alpha = 0.05$, the SPA method was conditionally invalid around 43% of possible realizations of $\sum_i Y_i$, and at significance level $\alpha = 5 \cdot 10^{-5}$, the SPA method was conditionally invalid around 39% of situations. As opposed to the normal approximation method where invalid tests clustered towards skewed response distributions, the SPA method fluctuates relatively evenly between conditionally valid and conditionally invalid as the number of cases v increases for both significance levels 0.05 and $5 \cdot 10^{-5}$. Therefore, the test is approximately equally good at any μ (Figure 2). Furthermore, the absolute differences in type I error rate control improves as the significance level decreases. This observation has a simple explanation. For some data sets, the SPA method yields the same critical region as the exact test, while at times the critical region is shifted by as little as one unit ($c_U^* = c_U - 1$ or $c_L^* = c_L + 1$). At a significance level of $\alpha = 0.05$, this shift can result in a substantial inflation in type I error rates, while at small significance levels, point probabilities are of such small magnitudes that the shift is less notable. As critical regions oscillate between correct and slightly shifted, conditional type I error rates oscillate above and below α, and averaging out to produce an overall type I error rate $\approx \alpha$.

5.2 Simulations of genetic association studies with an imbalanced response

The purpose of the following simulation study is to compare methods in a setting resembling a genome-wide association study with an imbalanced response, for which exact tests are not available. The simu-

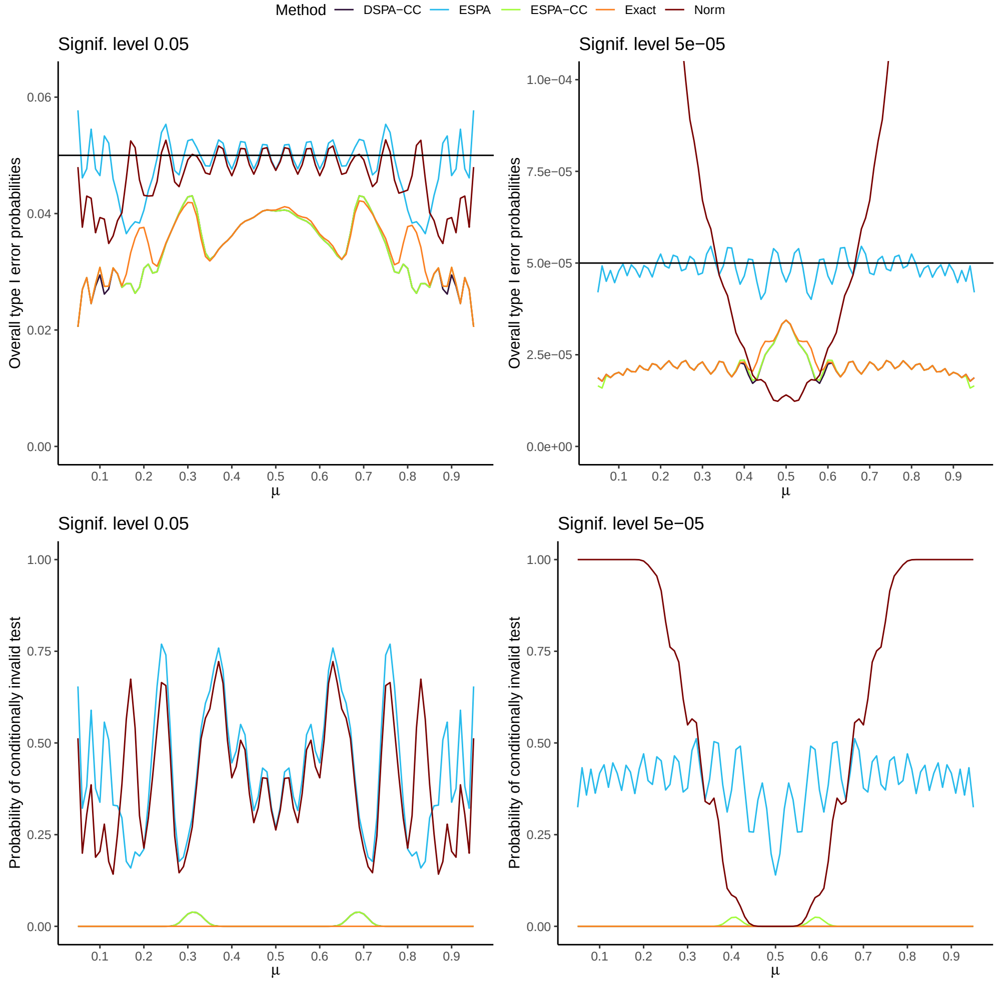


Figure 2: Exact overall type I error probabilities as well as probability of conditionally invalid tests for the different approximations methods for the distribution of the score test statistic, for different values of μ in the intercept model. We compare with the exact test using the known distribution of the score test statistic.

lation set-up is motivated by Dey et al. (2017) by conditioning on the number of cases, and we estimate the type I error rate *conditional* on the number of cases. The sample size considered is $n = 20000$, with case proportion 2% and 0.2%. We consider the logistic regression model

$$\text{logit}(\mu_i) = \beta_0 + x_{i,1} + x_{i,2} + \gamma g_i,$$

with $X_1 \sim \text{Bernoulli}(0.5)$, $X_2 \sim N(0, 1)$ and $G \sim \text{binom}(2, \text{MAF})$ with the MAF taking the values 0.05, 0.005, 0.0005 and 0.00025. Since we are evaluating validity of tests, we set $\gamma = 0$. Finally, we set $\beta_0 = -5.6$ such that the disease prevalence is 1% in the population.

The covariates $x_{i,1}$ and $x_{i,2}$ are sampled conditionally on their respective phenotype value y_i, while the genotype value is sampled independently of this under the null hypothesis. See Supplementary File for details. This ensures that the number of cases is equal for all simulations. For each set of case proportion and MAF, we simulate 10^9 data sets and record the number of times the null hypothesis is rejected at the $\alpha = 5 \cdot 10^{-8}$ significance level when using (1) the double saddlepoint approximation with continuity correction (DSPA-CC), (2) the continuity-corrected univariate saddlepoint approximation based on the efficient score (ESPA-CC), and (3) the continuous univariate saddlepoint approximation of the efficient score (ESPA). The resulting empirical type I error rates are presented in Figure 3, along with 95% Clopper-Pearson confidence intervals.

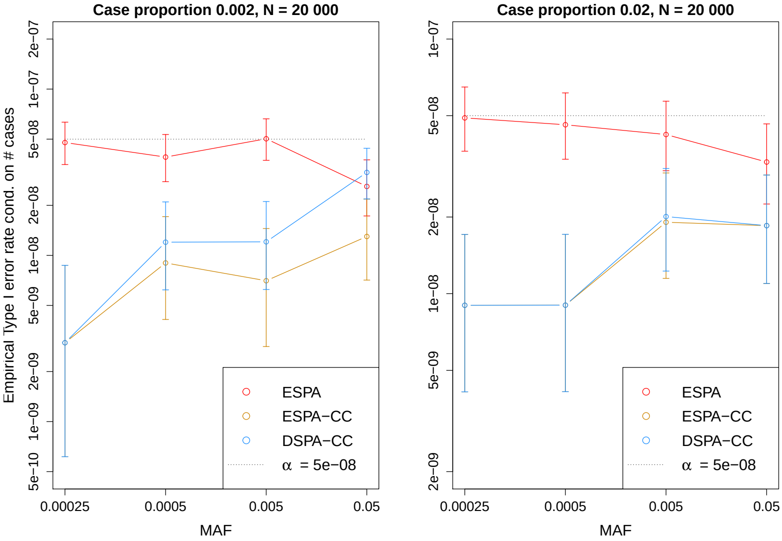


Figure 3: Approximated expected type I error rates - conditional on the number of cases - for ESPA, ESPA-CC and DSPA-CC from simulations with case proportions 0.02 and 0.002, and for small MAFs when nuisance covariates are included.

The simulation results closely follow the observations made in the previous section. The DSPA-CC and ESPA-CC are conservative (overall probability of type I error $< \alpha$), while the type I error rate of the ESPA method is $\approx \alpha$. The results are comparable with the pattern for conditionally invalid tests in Figure 2, specifically for the small case proportion, in that we sense a large fluctuation in the probability of invalid tests for ESPA, while both ESPA-CC and DSPA-CC have a small probability of invalid test, which is decreasing for decreasing MAF. We also observe that the type I error rate, conditional on the number of cases, for EPSA is increasing for decreasing MAF. The simulation study with case proportion 0.002 serves to illustrate deviations between the DSPA-CC and ESPA-CC method, and we observe that the ESPA-CC is somewhat more conservative in this setting.

6 Application to UK biobank data

We consider a recent GWAS in the UK Biobank with motivation from Rogne et al. (2021). The phenotype of interest is skin and soft tissue infections (SSTIs), and individuals are defined as cases if they have been hospitalized with main ICD-10 codes A46 (erysipelas), L03 (cellulitis and acute lymphangitis), or M72.6 (necrotizing fasciitis) in the period between the end of the recruitment period (2010-10-01) and April 2017 (2017-03-31). Individuals who had reported ICD-10 codes, or corresponding ICD-9 codes (035 and 729.4), before 2010-10-01 are removed as well as individuals with date of death reported after 2010-10-01 in the death register (see Data-Field 40000 in the UK Biobank data). As nuisance covariates we include age when attended assessment centre, genetic sex, and four principal components. To avoid complexities due to cryptic relatedness we only include unrelated individuals reported as White British (achieved through Data-Field 22006 and 22020 in UK Biobank). The principal components are calculated using

EIGENSOFT (version 6.1.4) SmartPCA (Patterson et al., 2006; Price et al., 2006). Only directly genotyped SNPs are considered, and phenotype-independent quality control of the genetic data is completed using PLINK1.9, with details given in the Supplementary File. This results in a total of 293 964 individuals and 529 024 SNPs with 2051 individuals defined as cases and 291 913 controls, resulting in a case proportion of 0.7 %. All SNPs are first investigated by computing p-values using the normal approximation to the score test statistic. As this test is proven to be too optimistic, SNPs with p-values less than $\alpha = 5 \cdot 10^{-5}$ are investigated more thoroughly by computing p-values using the DSPA-CC and ESPA-CC methods as implemented by us, as well the SPA-test of Dey et al. (2017), denoted ESPA. Dey et al. (2017) also propose a computationally more efficient approximation to their SPA-test by essentially assuming that the nuisance covariates are balanced. In a double saddlepoint setting, this assumption may be generalized to argue that the score vector $\boldsymbol{U_\beta}$ approximately has a multivariate normal distribution under the null hypothesis. Taking a similar approach to Dey et al. (2017), we may partition the joint CDF of $\boldsymbol{U_\beta}$ and U_γ into a sum over all individuals with genotype value $g_i > 0$ and those with $g_i = 0$. For the latter sub-sample, the CGF simplifies to a CGF of the score vector $\boldsymbol{U^*_\beta}$ including individuals with $g_i = 0$. Assuming that also $\boldsymbol{U^*_\beta}$ is normal, this part of the joint CGF may be replaced by a normal CGF, and by pre-computing the variance of $\boldsymbol{U^*_\beta}$, an approximated double saddlepoint method may be computed based only on the sub-sample individuals with genotypes $g_i > 0$. Details may be found in the Supplementary File. For comparative purposes, we also compute p-values based on the fastSPA method of Dey et al. (2017) and our similar fastDSPA-CC approach.

Test results for the SNPs with the smallest normal-approximated p-values are given in Table 1. In this setting, we no longer know whether the null hypothesis is true or not for each variant. However, we expect only a tiny proportion of all variants where the null hypothesis is false. Even though no SNPs reached the significance level $\alpha = 5 \cdot 10^{-8}$, we see a pattern similar to the results for the intercept model and our simulation results. The normal approximation is the most optimistic, followed by ESPA and fastSPA tests. The DSPA-CC test is more conservative, while the most conservative test is ESPA-CC. The fastDSPA-CC is slightly less conservative than DSPA-CC. The greatest difference between test results is observed for the SNP with a small minor allele frequency (rs113113104, MAF = 0.03). The difference between the p-values reduces for increasing MAFs. For the SNP rs566530 with MAF = 0.48, the SPA test gives a smaller p-value than the normal approximation, while the other methods give consistently larger p-values.

Table 1: The common variants with the smallest computed p-values using normal approximation to the score test statistic for the GWAS of skin and soft tissue infections. Alternative p-value computations are included for comparison.

SNP	**CHR**	**MAF**	**Norm. apx.**	**ESPA**	**fastSPA**	**SPA-CC**	**DSPA-CC**	**fastDSPA-CC**
rs113113104	6	0.03	2.39e-07	5.97e-07	6.04e-07	7.27e-07	7.10e-07	6.52e-07
rs6551253	3	0.28	8.38e-06	8.47e-06	8.78e-06	9.18e-06	9.00e-06	8.92e-06
rs78404737	2	0.10	8.50e-06	9.63e-06	9.78e-06	1.08e-05	1.06e-05	1.00e-05
rs78696065	7	0.02	8.80e-06	1.54e-05	1.55e-05	1.89e-05	1.87e-05	1.75e-05
rs479947	6	0.11	1.19e-05	1.29e-05	1.33e-05	1.44e-05	1.42e-05	1.35e-05
rs566530	6	0.48	1.46e-05	1.40e-05	1.48e-05	1.50e-05	1.47e-05	1.48e-05
rs56355912	10	0.03	1.51e-05	2.16e-05	2.16e-05	2.57e-05	2.54e-05	2.38e-05
rs72733294	5	0.36	1.58e-05	1.60e-05	1.60e-05	1.72e-05	1.69e-05	1.69e-05
rs11074743	16	0.40	1.69e-05	1.68e-05	1.71e-05	1.80e-05	1.77e-05	1.77e-05
rs1562963	11	0.07	2.02e-05	1.99e-05	2.33e-05	2.26e-05	2.23e-05	2.13e-05

6.1 Rare variants

The difference between the methods becomes even larger when investigating rare variants. We consider the UK Biobank exome sequence data consisting of 45 596 unrelated individuals of European origin. We limit ourselves to White British individuals using the same requirements for the definition of SSTIs as for the common variants. This results in a total number of 30 210 individuals to investigate with 210 individuals defined as cases, once again leading to a case proportion of about 0.7 %. See the Supplementary File for further information about quality control. The principal components are computed as for the common variants analysis, however separately on these 30 210 individuals. We will in addition only consider chromosome 6 as well as rare variants with a minimum minor allele count (MAC) equal to 3. The results are given in Table 2.

Table 2: The rare variants with the smallest computed p-values using normal approximation to the score test statistic for the GWAS of skin and soft tissue infections. Alternative p-value computations are included for comparison.

SNP	**CHR**	**MAC**	**Norm. apx.**	**ESPA**	**fastSPA**	**ESPA-CC**	**DSPA-CC**	**fastDSPA-CC**
6:26045407:G:A	6	4	2.07e-36	4.31e-05	4.31e-05	2.2e-04	2.2e-04	2.2e-04
6:41097421:T:C	6	4	2.21e-32	4.92e-05	4.92e-05	2.6e-04	2.6e-04	2.5e-04
6:24852645:G:T	6	4	1.37e-25	8.93e-05	8.93e-05	4.4e-04	4.3e-04	4.2e-04
6:31772925:C:A	6	5	6.36e-23	1.3e-04	1.3e-04	6.0e-04	6.0e-04	5.8e-04
6:20402579:C:T	6	3	4.19e-22	0.0020	0.0020	0.010	0.010	0.010
6:132588925:C:T	6	6	8.78e-22	1.5e-04	1.5e-04	6.9e-04	6.9e-04	6.7e-04
6:17675831:G:A	6	3	8.94e-22	0.0020	0.0020	0.010	0.010	0.010
6:110960684:T:G	6	3	2.05e-21	0.0017	0.0017	0.0049	0.0049	0.0049
6:7894854:T:C	6	16	1.88e-20	3.07e-05	3.073e-05	1.2e-04	1.2e-04	1.0e-04
6:148514044:G:T	6	3	1.94e-20	0.0022	0.0022	0.011	0.011	0.011

It is clear that the normal approximation to the score test statistic is very inaccurate in this setting. However, we also see that the difference between ESPA and the other saddlepoint approximations with continuity correction differ in about one order of magnitude. As a result, we expect the importance of the continuity correction to be most consequential for rare variants. Another observation is that ESPA-CC and DSPA-CC are practically identical in this case. We also see that the speed-up approximation methods are more accurate which can be explained by observing that the accuracy of the multivariate normal approximation of $\mathbf{U}^*_{\boldsymbol{\beta}}$ in fastDSPA-CC, depends on the number of individuals with $g_i = 0$, which increases for decreasing MACs. The same applies for the approximation of the corresponding normal distribution in fastSPA.

7 Discussion

We have investigated different saddlepoint approximations for GWAS with binary phenotypes in order to achieve valid p-values. We have shown how the saddlepoint approximation introduced in Dey et al. (2017) can be interpreted as a two-stage procedure in which one first applies the efficient score to approximate the conditional score test statistic as an unconditional statistic, and then performs single-saddlepoint approximation. We further show how to apply the double saddlepoint approximation to directly approximate the conditional score test statistic.

We distinguish between conditional and overall type I error rate. Taking into account both these measures, we conclude that continuity-corrected saddlepoint approximations are most appropriate in this setting. The continuity-corrected double saddlepoint approximation, DSPA-CC, and single-saddlepoint approximation, ESPA-CC, using the efficient score are both considered to perform well, however there are situations in which ESPA-CC is somewhat more conservative than DSPA-CC, indicating DSPA-CC to be somewhat more powerful.

There are additional continuity correction variants, and the one used here is called the *second continuity correction*. A first and a third continuity correction are alternatives (Butler, 2007), and specifically the first continuity correction was also investigated with very similar results as when using the second continuity correction, however slightly more inaccurate when considering the intercept model, see Supplementary File. An alternative saddlepoint approximation to the CDF of a random variable is the one introduced in Lugannani and Rice (1980). This approximation gives the same results as the approximation by Barndorff-Nielsen (1990) in most situations. However, we observed in simulations that when the case proportion and MAF approaches zero, the approximation by Lugananni and Rice is inaccurate, see Supplementary File. See for instance Booth and Wood (1995) for similar observations in a different application.

Consider the case where one wants to include imputed SNPs. For most imputation methods, the output for each imputed SNP is a probability that the minor allele count is equal to 0, 1 or 2, denoted p_0, p_1 and p_2. Then one must be aware of the fact that when the imputed genotype is set to be the expected minor allele count, $p_1 + 2p_2$, the score test statistic will no longer have a lattice distribution, and so continuity correction does no longer apply. However, to account for imputed SNPs in our method one can instead set the imputed minor allele count to be equal to the most likely allele count according to the imputation method.

Single-variant tests on rare variants are often low-powered, and therefore several region-based tests including several SNPs in the same genetic region have been proposed to gain power. However, many of these methods again rely on single-variant tests as building blocks, among them SKAT and ACAT (Liu et al., 2019; Wu et al., 2011). It is therefore essential that the single-variant tests are sufficiently accurate. How the insights into the score test statistic introduced in this work would impact region-based tests, could be the topic of future research.

8 Acknowledgements

This research was supported by the Norwegian Research Council grant 272402 (PhD Scholarships at SINTEF) as well the funding for research stays abroad for doctoral and postdoctoral fellows financed by the Norwegian Research Council. The research has been conducted using the UK Biobank Resource under Application Number 32285. We thank the Yale Center for Research Computing for guidance and use of the research computing infrastructure. We thank the The Gemini Center for Sepsis Research for establishing cooperation with Yale School of Public Health.

9 Code availability

Source code is available at https://github.com/palVJ/SaddlePointApproxInBinaryGWAS.

A Proofs of Observations 1–3

Proof of Observation 1. When $g_i \in (0, 1, 2)$, we note that $\boldsymbol{g}^{\mathrm{T}}\boldsymbol{Y}$ is an integer and $\boldsymbol{g}^{\mathrm{T}}\boldsymbol{\mu}$ a constant, so that $U_\gamma = \boldsymbol{g}^{\mathrm{T}}\boldsymbol{Y} - \boldsymbol{g}^{\mathrm{T}}\boldsymbol{\mu}$ has support on a subset of a lattice with step 1. The minimum is obtained for $\boldsymbol{Y} = \boldsymbol{0}$ and the maximum for $\boldsymbol{Y} = \boldsymbol{1}$ (a vector of ones), and the result follows. □

Proof of Observation 2. We assume throughout the proof that the null hypothesis is true, $\gamma = 0$. Denote by V_j the sum of responses Y_i among individuals with genotype $g_i = j,\ \ j = 0,\ 1,\ 2$, and let $V = V_0 + V_1 + V_2 = \sum_{i=1}^{n} Y_i$ be the total sum of responses. With this notation, $U_\gamma = V_1 + 2V_2 - (n_1 + 2n_2)\mu$, and $U_\beta = V - n\mu$, so that the condition $U_\beta = 0$ is equivalent to $V = n\mu$.

The V_j are independent, and V_j is binomially distributed with parameters n_j and $\mu,\ \ j = 0,\ 1,\ 2$, and V is binomially distributed with parameters n and μ. Assume that $v_0 + v_1 + v_2 = n\mu$ with v_j in the support of V_j. Then

$$\begin{aligned}P(V_0 = v_0, V_1 = v_1, V_2 = v_2 \mid V = n\mu) &= \frac{P(V_0 = v_0)P(V_1 = v_1)P(V_2 = v_2)}{P(V = n\mu)} \\ &= \frac{\binom{n_0}{v_0}\mu^{v_0}(1-\mu)^{n_0 - v_0}\binom{n_1}{v_1}\mu^{v_1}(1-\mu)^{n_1 - v_1}\binom{n_2}{v_2}\mu^{v_2}(1-\mu)^{n_2 - v_2}}{\binom{n}{n\mu}\mu^{n\mu}(1-\mu)^{n - n\mu}} = \frac{\binom{n_0}{v_0}\binom{n_1}{v_1}\binom{n_2}{v_2}}{\binom{n}{n\mu}},\end{aligned}$$

a trivariate hypergeometric probability.

Now, $P(U_\gamma = u \mid U_\beta = 0) = P\big(V_1 + 2V_2 = u^* \bigm| V = n\mu\big)$ can be found by summing the above probabilities over $(v_0, v_1, v_2) \in S$. This gives the first sum of the Observation. The more explicit second version of the sum is obtained by solving the two equations in the definition of S for v_0 and v_1 in terms of $k = v_2$. The limits of the sum is determined by the inequalities in the definition of S. □

Proof of Observation 3. We assume throughout the proof that the null hypothesis is true, $\gamma = 0$. Denote by V_j the sum of responses Y_i among individuals with $x_i = 0$ and genotype $g_i = j,\ \ j = 0,\ 1,\ 2$, and let $V = V_0 + V_1 + V_2$. Define similar sums W_j and W for individuals with $x_i = 1$. With this notation, $U_\gamma = V_1 + 2V_2 - (l_1 + 2l_2)\mu_0 + W_1 + 2W_2 - (m_1 + 2m_2)\mu_1$, and $\boldsymbol{U}_{\boldsymbol{\beta}}^{\mathrm{T}} = \big(V + W - l\mu_0 - m\mu_1 \quad W - m\mu_1\big)$, so that the condition $\boldsymbol{U}_{\boldsymbol{\beta}} = \boldsymbol{0}$ is equivalent to $V = l\mu_0$ and $W = m\mu_1$.

All the V_j and W_j are independent, and V_j is binomially distributed with parameters l_j and μ_0, and W_j with parameters m_j and μ_1, $j = 0, 1, 2$. As in the proof of Observation 2, the conditional point probabilites of (V_0, V_1, V_2) given $V = l\mu_0$ and (W_0, W_1, W_2) given $W = m\mu_1$ are trivariate hypergeometric probabilities, and by independence of the two triples, the conditional joint point probability is the product of the two. Then $P(U_\gamma = u \mid \boldsymbol{U_\beta} = \mathbf{0})$ can be found by summing those probabilities over $\boldsymbol{s} \in S$. □

B Support of the conditional score test statistic

Consider the score test statistic of U_γ conditional on $\boldsymbol{U_\beta} = \mathbf{0}$, given by $\boldsymbol{g}^T(\boldsymbol{Y} - \hat{\boldsymbol{\mu}})$. We have $-\hat{\boldsymbol{\mu}} \leq \boldsymbol{Y} - \hat{\boldsymbol{\mu}} \leq \mathbf{1} - \hat{\boldsymbol{\mu}}$ (elementwise inequalities), where $\mathbf{1}$ is a vector of ones. Since all $g_i \geq 0$, premultiplying the inequalities with $\boldsymbol{g}^{\mathrm{T}}$ gives bounds on the support of $\boldsymbol{g}^T(\boldsymbol{Y} - \hat{\boldsymbol{\mu}})$:

$$-\boldsymbol{g}^{\mathrm{T}}\hat{\boldsymbol{\mu}} \leq U_\gamma \leq \boldsymbol{g}^{\mathrm{T}}(\mathbf{1} - \hat{\boldsymbol{\mu}}). \tag{11}$$

The first equality holds when $\boldsymbol{g}^{\mathrm{T}}\boldsymbol{Y} = 0$ and the second when $\boldsymbol{g}^{\mathrm{T}}\boldsymbol{Y} = \boldsymbol{g}^{\mathrm{T}}\mathbf{1}$. However, this combination is not achievable if it does not satisfy $\boldsymbol{U_\beta} = \boldsymbol{X}^T(\boldsymbol{Y} - \hat{\boldsymbol{\mu}}) = \mathbf{0}$. Specifically, the minimal and maximal achievable values of the conditional score test statistic is given by the constraint optimization problems:

$$\begin{aligned} \min(U_\gamma) = \min_{\boldsymbol{y}} \quad & \boldsymbol{g}^T(\boldsymbol{y} - \hat{\boldsymbol{\mu}}) \\ \text{such that} \quad & X^T(\boldsymbol{y} - \hat{\boldsymbol{\mu}}) = \mathbf{0}, \end{aligned}$$

and

$$\begin{aligned} \max(U_\gamma) = \max_{\boldsymbol{y}} \quad & \boldsymbol{g}^T(\boldsymbol{y} - \hat{\boldsymbol{\mu}}) \\ \text{such that} \quad & X^T(\boldsymbol{y} - \hat{\boldsymbol{\mu}}) = \mathbf{0}. \end{aligned}$$

As an example, consider the intercept model with $n = 1000$ and $\boldsymbol{g}$ as in Section 5.1 with $n_0 = 980$, $n_1 = 20$ and $n_2 = 0$ as well as the observation $\sum_{i=1}^{1000} Y_i = 10$. Then $\hat{\mu}_i = 10/1000 = 0.01$ satisfies $U_{\beta_0} = \sum_{i=1}^{1000}(Y_i - \mu_i) = 0$. Then the minimum achievable value is indeed $\min(U_\gamma) = -\boldsymbol{g}^T\hat{\boldsymbol{\mu}} = -0.2$, since we may have a combination where $Y_i = 0$ for all $g_i > 0$, and still get $\sum_{i=1}^{1000} Y_i = 10$. However, $\max(U_\gamma) = 10 - \boldsymbol{g}^T\hat{\boldsymbol{\mu}} = 9.8$ since $\boldsymbol{g}^T\boldsymbol{Y}$ can be no larger than the combinations where $g_i = 1$ for all $Y_i = 1$, which can only occur ten times in order to satisfy $\sum_{i=1}^{1000} Y_i = 10$.

C Solution to $\nabla_{\boldsymbol{t_\beta}} K_{\boldsymbol{\beta}}(\tilde{\boldsymbol{t}}_{\boldsymbol{\beta}}) = \mathbf{0}$

Given the marginal cumulant generating function of $\boldsymbol{U_\beta}$, defined by $K_{\boldsymbol{\beta}}(\boldsymbol{t_\beta})$ (a function of d variables) with

$$K_{\boldsymbol{\beta}}(\boldsymbol{t_\beta}) = \sum_{i=1}^{n} \ln(1 - \mu_i + \mu_i \exp(\mathbf{x}_i^T \boldsymbol{t_\beta})) - \boldsymbol{t}_{\boldsymbol{\beta}}^T X^T \boldsymbol{\mu}, \tag{12}$$

and corresponding gradient

$$\nabla_{\boldsymbol{t_\beta}} K_{\boldsymbol{\beta}}(\boldsymbol{t_\beta}) = \sum_{i=1}^{n} \mu_i \mathbf{x}_i \left(\frac{1}{(1 - \mu_i)\exp(-\mathbf{x}_i^T \boldsymbol{t_\beta}) + \mu_i} - 1 \right). \tag{13}$$

First, one can easily observe that $\tilde{\boldsymbol{t}}_{\boldsymbol{\beta}} = \mathbf{0}$ is a solution to $\nabla_{\boldsymbol{t_\beta}} K_{\boldsymbol{\beta}}(\boldsymbol{t_\beta}) = \mathbf{0}$. Second, if one can prove that the CGF is a convex function, then $\tilde{\boldsymbol{t}}_{\boldsymbol{\beta}} = \mathbf{0}$ is a unique solution to $\nabla_{\boldsymbol{t_\beta}} K_{\boldsymbol{\beta}}(\boldsymbol{t_\beta}) = \mathbf{0}$.

Proof. In fact, convexity of a cumulant generating function with *any* random variable $\boldsymbol{U}$, $K(\boldsymbol{t}) = \ln E(e^{\boldsymbol{t}^T\boldsymbol{U}})$, in general follows from the Hölder inequality, $E(|X|^c|Y|^{1-c}) \leq (E|X|)^c(E|Y|)^{1-c}$ for all

c in $(0, 1)$, where X and Y are random variables. A function f is convex if $f(c\boldsymbol{t}_1 + (1 - c)\boldsymbol{t}_2) \leq cf(\boldsymbol{t}_1) + (1 - c)f(\boldsymbol{t}_2)$ for all c in $(0, 1)$. Now,

$$\begin{aligned} K(c\boldsymbol{t}_1 + (1-c)\boldsymbol{t}_2) &= \ln E e^{(c\boldsymbol{t}_1+(1-c)\boldsymbol{t}_2)^{\mathrm{T}}\boldsymbol{U}} = \ln E\big(e^{c\boldsymbol{t}_1^{\mathrm{T}}\boldsymbol{U}} e^{(1-c)\boldsymbol{t}_2^{\mathrm{T}}\boldsymbol{U}}\big) \\ &\leq \ln\big(\big(Ee^{\boldsymbol{t}_1^{\mathrm{T}}\boldsymbol{U}}\big)^c \big(Ee^{\boldsymbol{t}_2^{\mathrm{T}}\boldsymbol{U}}\big)^{1-c}\big) = c\ln Ee^{\boldsymbol{t}_1^{\mathrm{T}}\boldsymbol{U}} + (1-c)\ln Ee^{\boldsymbol{t}_2^{\mathrm{T}}\boldsymbol{U}} \\ &= cK(\boldsymbol{t}_1) + (1-c)K(\boldsymbol{t}_2), \end{aligned}$$

showing that K is convex. □

References

Barndorff-Nielsen, O. E. (1990). Approximate Interval Probabilities. *Journal of the Royal Statistical Society. Series B (Methodological)*, *52*(3), 485–496.

Bickel, P. J., Klaassen, C. A., Ritov, Y., & Wellner, J. A. (1993). *Efficient and adaptive estimation for semiparametric models* (Vol. 4). Johns Hopkins University Press Baltimore.

Booth, J. G., & Wood, A. T. A. (1995). An example in which the Lugannani-Rice saddlepoint formula fails. *Statistics & Probability Letters*, *23*(1), 53–61.

Butler, R. W. (2007). *Saddlepoint approximations with applications*. Cambridge University Press.

Cox, D. R., & Reid, N. (1987). Parameter orthogonality and approximate conditional inference. *Journal of the Royal Statistical Society: Series B (Methodological)*, *49*(1), 1–18.

Dey, R., Schmidt, E. M., Abecasis, G. R., & Lee, S. (2017). A fast and accurate algorithm to test for binary phenotypes and its application to PheWAS. *American Journal of Human Genetics*, *101*(1), 37–49.

Firth, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika*, *80*(1), 27–38.

Jannot, A.-S., Ehret, G., & Perneger, T. (2015). $P < 5 \times 10^{-8}$ has emerged as a standard of statistical significance for genome-wide association studies. *Journal of clinical epidemiology*, *68*(4), 460–465.

Lindsey, J. K. (1996). *Parametric statistical inference*. Oxford University Press.

Liu, Y., Chen, S., Li, Z., Morrison, A. C., Boerwinkle, E., & Lin, X. (2019). ACAT: A Fast and Powerful p Value Combination Method for Rare-Variant Analysis in Sequencing Studies. *The American Journal of Human Genetics*, *104*(3), 410–421.

Lugannani, R., & Rice, S. (1980). Saddle point approximation for the distribution of the sum of independent random variables. *Advances in Applied Probability*, *12*(2), 475–490. https://doi.org/10.2307/1426607

Ma, C., Blackwell, T., Boehnke, M., & Scott, L. J. (2013). Recommended joint and meta-analysis strategies for case-control association testing of single low-count variants. *Genetic epidemiology*, *37*(6), 539–550.

Patterson, N., Price, A. L., & Reich, D. (2006). Population Structure and Eigenanalysis. *PLOS Genetics*, *2*(12), e190.

Price, A. L., Patterson, N. J., Plenge, R. M., Weinblatt, M. E., Shadick, N. A., & Reich, D. (2006). Principal components analysis corrects for stratification in genome-wide association studies. *Nature Genetics*, *38*(8), 904–909.

Rogne, T., Liyanarachi, K. V., Rasheed, H., Thomas, L. F., Flatby, H. M., Stenvik, J., Løset, M., Gill, D., Burgess, S., Willer, C. J., Hveem, K., Åsvold, B. O., Brumpton, B. M., DeWan, A. T., Solligård, E., & Damås, J. K. (2021). GWAS Identifies LINC01184/SLC12A2 as a Risk Locus for Skin and Soft Tissue Infections. *J Invest Dermatol.*

Skovgaard, I. M. (1987). Saddlepoint expansions for conditional distributions. *Journal of Applied Probability*, *24*(4), 875–887.

Smyth, G. K. (2003). Pearson's goodness of fit statistic as a score test statistic. *Lecture notes-monograph series*, 115–126.

Waterman, R. P., & Lindsay, B. G. (1996). A simple and accurate method for approximate conditional inference applied to exponential family models. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 177–188.

Wu, M. C., Lee, S., Cai, T., Li, Y., Boehnke, M., & Lin, X. (2011). Rare-Variant Association Testing for Sequencing Data with the Sequence Kernel Association Test. *American Journal of Human Genetics*, *89*(1), 82–93.

Saddlepoint approximations in binary genome-wide association studies
Supplementary File

Pål Vegard Johnsen[1,2], Øyvind Bakke[2], Thea Bjørnland[2], Andrew Thomas DeWan[3], and Mette Langaas[2]

[1]SINTEF Digital, Oslo, Norway
[2]Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway
[3]Department of Chronic Disease Epidemiology and Center for Perinatal, Pediatric and Environmental Epidemiology, Yale School of Public Health

1 Quality assessment of UK Biobank Genetic Data

1.1 Common variants

Analyses were limited to autosomal variants covered by both genotype arrays used over the course of the study and that passed the batch-level quality control. SNPs were included if the call rate was above 99%, the Hardy-Weinberg equilibrium p-value was less than $5 \cdot 10^{-8}$, and the minor allele frequency was larger than 1%. 529 024 SNPs passed these filters.

Individuals were removed if the genetic and reported sex did not match and if the sex chromosomes were not XX or XY. Outliers in heterozygosity and missing rates were removed. The analyses were limited to those identified as Caucasian through the UK Biobank's PCA analysis (field 22006). All individuals had an individual call rate larger than 99%. 366 752 individuals passed these filters. Individuals were removed if the genetic and reported sex did not match and if the sex chromosomes were not XX or XY.

1.2 Rare variants

Analyses were limited to autosomal variants at chromosome 6. Details of quality assessment of the sequenced exomes from 49 960 UKB participants is given in Van Hout et al. (2019). For further quality assessment, out of these 49 960 participants, the analysis were limited to those identified as Caucasian through the UK Biobank's PCA analysis (field 22006). Individuals were removed if the genetic and reported sex did not match and if the sex chromosomes were not XX or XY. The SNPs had a MAF less than 0.01, but a MAC larger than two. All SNPs had a missing rate less than 0.01, and all individuals had an individual call rate larger than 99%.

2 Approximating the double saddlepoint method by a normal approximation to $\boldsymbol{U}_\beta$

By a double saddlepoint method we may, under the null, estimate

$$P(U_\gamma = u_\gamma | \boldsymbol{U_\beta} = \mathbf{0}) = \frac{f(\mathbf{0}, u_\gamma)}{f_{\boldsymbol{\beta}}(\mathbf{0})}, \tag{1}$$

by using saddlepoint techniques to approximate the joint distribution f of $\boldsymbol{U_\beta}$ and U_γ at $\boldsymbol{U_\beta} = \mathbf{0}$, and the marginal distribution $f_{\boldsymbol{\beta}}$ of $\boldsymbol{U_\beta}$ at $\boldsymbol{U_\beta} = \mathbf{0}$.

2.1 The joint cumulant generating function

Let $\boldsymbol{t}$ be a vector of dimension $d+1$, which we partition into the vector $\boldsymbol{t}_\beta$ of dimension d and the scalar t_γ. Since $Y_i \sim \text{binomial}(\mu_i)$, the CGF of $\boldsymbol{U}$ may then be expressed as

$$\begin{aligned} K(\boldsymbol{t}) = K(\boldsymbol{t}_{\boldsymbol{\beta}}, t_\gamma) &= \sum_{i=1}^{n} \ln\left(1 - \mu_i + \mu_i \exp(g_i t_\gamma + \boldsymbol{t}_{\boldsymbol{\beta}}^T \boldsymbol{x}_i)\right) \\ &\quad - t_\gamma \sum_{i=1}^{n} g_i \mu_i - \boldsymbol{t}_{\boldsymbol{\beta}}^T \sum_{i=1}^{n} \boldsymbol{x}_i \mu_i. \end{aligned} \tag{2}$$

Derivatives of $K(\boldsymbol{t}_\beta, t_\gamma)$ with respect to $\boldsymbol{t}_\beta$ and t_γ, denoted $\nabla_{\boldsymbol{t}_{\boldsymbol{\beta}}} K(\boldsymbol{t}_\beta, t_\gamma)$ and $\frac{\partial}{\partial t_\gamma} K(\boldsymbol{t}_\beta, t_\gamma)$ respectively, are

$$\nabla_{\boldsymbol{t}_{\boldsymbol{\beta}}} K(\boldsymbol{t}_{\boldsymbol{\beta}}, t_\gamma) = \sum_{i=1}^{n} \mu_i \left(\frac{\exp(g_i t_\gamma + \boldsymbol{t}_{\boldsymbol{\beta}}^T \boldsymbol{x}_i)}{\left(1 - \mu_i + \mu_i \exp(g_i t_\gamma + \boldsymbol{t}_{\boldsymbol{\beta}}^T \boldsymbol{x}_i)\right)} - 1 \right) \boldsymbol{x}_i,$$

and

$$\frac{\partial}{\partial t_\gamma} K(\boldsymbol{t}_{\boldsymbol{\beta}}, t_\gamma) = \sum_{i=1}^{n} \mu_i \left(\frac{\exp(g_i t_\gamma + \boldsymbol{t}_{\boldsymbol{\beta}}^T \boldsymbol{x}_i)}{\left(1 - \mu_i + \mu_i \exp(g_i t_\gamma + \boldsymbol{t}_{\boldsymbol{\beta}}^T \boldsymbol{x}_i)\right)} - 1 \right) g_i,$$

so that the gradient of $K(\boldsymbol{t}_{\boldsymbol{\beta}}, t_\gamma)$, denoted $\nabla K(\boldsymbol{t}_{\boldsymbol{\beta}}, t_\gamma)$, may be expressed as

$$\nabla K(\boldsymbol{t}_{\boldsymbol{\beta}}, t_\gamma) = \begin{pmatrix} \nabla_{\boldsymbol{t}_{\boldsymbol{\beta}}} K(\boldsymbol{t}_{\boldsymbol{\beta}}, t_\gamma) \\ \frac{\partial}{\partial t_\gamma} K(\boldsymbol{t}_{\boldsymbol{\beta}}, t_\gamma) \end{pmatrix}.$$

Let $\boldsymbol{\theta} = \begin{pmatrix}\boldsymbol{\beta}^{\mathrm{T}} & \gamma\end{pmatrix}^{\mathrm{T}}$ denote the full parameter set, and define a diagonal matrix $M^{\boldsymbol{\theta}}$ with entries

$$M_{ii}^{\boldsymbol{\theta}} = \frac{\mu_i(1-\mu_i)\exp(-g_i t_\gamma - \boldsymbol{t}_{\boldsymbol{\beta}}^T \boldsymbol{x}_i)}{\left((1-\mu_i)\exp(-g_i t_\gamma - \boldsymbol{t}_{\boldsymbol{\beta}}^T \boldsymbol{x}_i) + \mu_i\right)^2}.$$

The Hessian of K, denoted $H(\boldsymbol{t})$, can be expressed as

$$H(\boldsymbol{t}) = \begin{bmatrix} \frac{\partial^2}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} K(\boldsymbol{t}_{\boldsymbol{\beta}}, t_\gamma) & \frac{\partial^2}{\partial \boldsymbol{\beta} \partial t_\gamma} K(\boldsymbol{t}_{\boldsymbol{\beta}}, t_\gamma) \\ \frac{\partial^2}{\partial t_\gamma \partial \boldsymbol{\beta}} K(\boldsymbol{t}_{\boldsymbol{\beta}}, t_\gamma) & \frac{\partial^2}{\partial t_\gamma^2} K(\boldsymbol{t}_{\boldsymbol{\beta}}, t_\gamma) \end{bmatrix} = \begin{bmatrix} X^T M \boldsymbol{X} & X^T M \boldsymbol{g} \\ \boldsymbol{g}^T M X & \boldsymbol{g}^T M \boldsymbol{g} \end{bmatrix}.$$

2.2 The marginal cumulant generating function

The cumulant generating function of $\boldsymbol{U}_{\boldsymbol{\beta}}$, denoted $K_{\boldsymbol{\beta}}(\boldsymbol{t}_{\boldsymbol{\beta}})$, is given by

$$K_{\boldsymbol{\beta}}(\boldsymbol{t}_{\boldsymbol{\beta}}) = \sum_{i=1}^{n} \ln\left(1 - \mu_i + \mu_i \exp\left(\boldsymbol{t}_{\boldsymbol{\beta}}^T \boldsymbol{x}_i\right)\right) - \sum_{i=1}^{n} \boldsymbol{t}_{\boldsymbol{\beta}}^T \boldsymbol{x}_i \mu_i. \tag{3}$$

The gradient, denoted $\nabla K_{\boldsymbol{\beta}}(\boldsymbol{t}_{\boldsymbol{\beta}})$, is

$$\nabla K_{\boldsymbol{\beta}}(\boldsymbol{t}_{\boldsymbol{\beta}}) = \sum_{i=1}^{n} \mu_i \left(\frac{\exp(\boldsymbol{t}_{\boldsymbol{\beta}}^T \boldsymbol{x}_i)}{\left(1 - \mu_i + \mu_i \exp(g_i t_\gamma + \boldsymbol{t}_{\boldsymbol{\beta}}^T \boldsymbol{x}_i)\right)} - 1 \right) \boldsymbol{x}_i,$$

and the Hessian, denoted $H_{\boldsymbol{\beta}}(\boldsymbol{t}_{\boldsymbol{\beta}})$, is

$$H_{\boldsymbol{\beta}}(\boldsymbol{t}_{\boldsymbol{\beta}}) = X^T M^{\boldsymbol{\beta}} X,$$

where $M^{\boldsymbol{\beta}}$ is a diagonal matrix with entries

$$M_{ii}^{\boldsymbol{\beta}} = \frac{\mu_i(1-\mu_i)\exp(-\boldsymbol{t}_{\boldsymbol{\beta}}^T \boldsymbol{x}_i)}{\left((1-\mu_i)\exp(-\boldsymbol{t}_{\boldsymbol{\beta}}^T \boldsymbol{x}_i) + \mu_i\right)^2}.$$

We note that

$$K(\boldsymbol{t}_{\boldsymbol{\beta}}, 0) = K_{\boldsymbol{\beta}}(\boldsymbol{t}_{\boldsymbol{\beta}}). \tag{4}$$

2.3 Double-saddlepoint approximation

The saddlepoint approximation of the probability distribution of the score vector $\boldsymbol{U}$ evaluated at $\boldsymbol{U_\beta} = 0$ is given by

$$\hat{f}(\mathbf{0}, u_\gamma) = (2\pi)^{-(d+1)/2} |H(\hat{\boldsymbol{t}})|^{-1/2} \exp\left\{K(\hat{\boldsymbol{t}}_{\boldsymbol{\beta}}, \hat{t}_\gamma) - \hat{t}_\gamma u_\gamma)\right\},$$

where $\begin{pmatrix}\hat{\boldsymbol{t}}_{\boldsymbol{\beta}}^{\mathrm{T}} & \hat{t}_\gamma\end{pmatrix}^{\mathrm{T}}$ is the $(d+1)$-dimensional saddlepoint that solves $K'(\hat{\boldsymbol{t}}_{\boldsymbol{\beta}}, \hat{t}_\gamma) = \begin{pmatrix}\mathbf{0}^{\mathrm{T}} & u_\gamma\end{pmatrix}^{\mathrm{T}}$. The saddlepoint approximation of the marginal distribution of $\boldsymbol{U_\beta}$, evaluated at $\boldsymbol{U_\beta} = \mathbf{0}$ is similarly

$$\hat{f}_{\boldsymbol{\beta}}(\mathbf{0}) = (2\pi)^{-d/2} |H_{\boldsymbol{\beta}}(\tilde{\boldsymbol{t}}_{\boldsymbol{\beta}})|^{-1/2} \exp\left\{K_{\boldsymbol{\beta}}(\tilde{\boldsymbol{t}}_{\boldsymbol{\beta}})\right\},$$

where $\tilde{\boldsymbol{t}}_{\boldsymbol{\beta}}$ is the d-dimensional saddlepoint that solves $\nabla K_{\boldsymbol{\beta}}(\boldsymbol{t}_{\boldsymbol{\beta}}) = \mathbf{0}$. We showed in Appendix B that $\tilde{\boldsymbol{t}}_{\boldsymbol{\beta}} = \mathbf{0}$, hence

$$\hat{f}_{\boldsymbol{\beta}}(\mathbf{0}) = (2\pi)^{-d/2} |H_{\boldsymbol{\beta}}(\mathbf{0})|^{-1/2} \exp\left\{K_{\boldsymbol{\beta}}(\mathbf{0})\right\}.$$

2.4 Speed-up algorithm

Starting with the joint CGF $K(\boldsymbol{t}_{\boldsymbol{\beta}}, t_\gamma)$ in Equation (2) we split the sum into the sets of individuals with and without minor alleles:

$$\begin{aligned} K(\boldsymbol{t}_{\boldsymbol{\beta}}, t_\gamma) = &\sum_{i=1}^{m} \ln\left(1 - \mu_i + \mu_i \exp(g_i t_\gamma + \boldsymbol{t}_{\boldsymbol{\beta}}^T \boldsymbol{x}_i)\right) - t_\gamma \sum_{i=1}^{m} g_i \mu_i - \boldsymbol{t}_{\boldsymbol{\beta}}^T \sum_{i=1}^{m} \boldsymbol{x}_i \mu_i \\ &+ \sum_{i=m+1}^{n} \ln\left(1 - \mu_i + \mu_i \exp(\boldsymbol{t}_{\boldsymbol{\beta}}^T \boldsymbol{x}_i)\right) - \boldsymbol{t}_{\boldsymbol{\beta}}^T \sum_{i=m+1}^{n} \boldsymbol{x}_i \mu_i. \end{aligned}$$

By comparing with Equation (3), the last two terms is in fact the part of the cumulant generating function of $\boldsymbol{U}_\beta$ restricted to individuals with $g_i = 0$, denoted $K_{\boldsymbol{\beta}}^*(\boldsymbol{t}_{\boldsymbol{\beta}})$. As discussed in Dey et al. (2017), if the non-genetic covariates are not particularly skewed, then a normal approximation to $\boldsymbol{U_\beta}$ may be accurate. If there are few individuals with $g_i > 0$, which is typically the case, this would imply that a normal approximation of $\boldsymbol{U}_{\boldsymbol{\beta}}^*$, the part of $\boldsymbol{U_\beta}$ with $g_i = 0$, may also be accurate. Therefore, let $X_{g_i=0}$, $\boldsymbol{Y}_{g_i=0}$ and $\boldsymbol{\mu}_{g_i=0}$ denote the part of X, $\boldsymbol{Y}$ and $\boldsymbol{\mu}$ with $g_i = 0$, and so $\boldsymbol{U}_{\boldsymbol{\beta}}^* = X_{g_i=0}^T(\boldsymbol{Y}_{g_i=0} - \boldsymbol{\mu}_{g_i=0})$ with $E(\boldsymbol{U}_{\boldsymbol{\beta}}^*) = \mathbf{0}$ and $\mathrm{Cov}(\boldsymbol{U}_{\boldsymbol{\beta}}^*) = X_{g_i=0}^T W_{g_i=0} X_{g_i=0}$, with $W_{g_i=0}$ the submatrix of the diagonal matrix W with entries $\mu_i(1-\mu_i)$ among those individuals with $g_i = 0$. By approximating $\boldsymbol{U}_{\boldsymbol{\beta}}^*$ to have a normal distribution, the approximation of the CGF of $U_{\boldsymbol{\beta}}^*$ is $K_{\boldsymbol{\beta}}^*(\boldsymbol{t}_{\boldsymbol{\beta}}) \approx \frac{1}{2}\boldsymbol{t}_{\boldsymbol{\beta}}^T \mathrm{Cov}(\boldsymbol{U}_{\boldsymbol{\beta}}^*)\boldsymbol{t}_{\boldsymbol{\beta}}$. And consequently, the original CGF may be approximated as

$$\begin{aligned} K(\boldsymbol{t}_{\boldsymbol{\beta}}, t_\gamma) \approx &\sum_{i=1}^{m} \ln\left(1 - \mu_i + \mu_i \exp(g_i t_\gamma + \boldsymbol{t}_{\boldsymbol{\beta}}^T \boldsymbol{x}_i)\right) - t_\gamma \sum_{i=1}^{m} g_i \mu_i - \boldsymbol{t}_{\boldsymbol{\beta}}^T \sum_{i=1}^{m} \boldsymbol{x}_i^T \mu_i \\ &+ \frac{1}{2}\boldsymbol{t}_{\boldsymbol{\beta}}^T \mathrm{Cov}(\boldsymbol{U}_{\boldsymbol{\beta}}^*)\boldsymbol{t}_{\boldsymbol{\beta}}. \end{aligned}$$

This approximation to the CGF does not represent any reasonable speed-up yet, since $\mathrm{Cov}(\boldsymbol{U}_{\boldsymbol{\beta}}^*)$ must be computed for each genetic variant and requires $O(n-m)$ calculations. However, we may express for each variant $\mathrm{Cov}(\boldsymbol{U}_{\boldsymbol{\beta}}^*) = \mathrm{Cov}(\boldsymbol{U_\beta}) - \mathrm{Cov}(\boldsymbol{U}_{\boldsymbol{\beta}}^\dagger)$, with $\mathrm{Cov}(\boldsymbol{U_\beta})$ the same for all variants, while $\boldsymbol{U}_{\boldsymbol{\beta}}^\dagger$ is the part of $\boldsymbol{U_\beta}$ with $g_i > 0$, and so $\mathrm{Cov}(\boldsymbol{U}_{\boldsymbol{\beta}}^\dagger) = X_{g_i>0}^T W_{g_i>0} X_{g_i>0}$. As $\mathrm{Cov}(\boldsymbol{U_\beta})$ can be precomputed for all variants, this requires $O(m)$ calculations. Hence, the approximation

$$\begin{aligned} K(\boldsymbol{t}_{\boldsymbol{\beta}}, t_\gamma) \approx &\sum_{i=1}^{m} \ln\left(1 - \mu_i + \mu_i \exp(g_i t_\gamma + \boldsymbol{t}_{\boldsymbol{\beta}}^T \boldsymbol{x}_i)\right) - t_\gamma \sum_{i=1}^{m} g_i \mu_i - \boldsymbol{t}_{\boldsymbol{\beta}}^T \sum_{i=1}^{m} \boldsymbol{x}_i^T \mu_i \\ &+ \frac{1}{2}\boldsymbol{t}_{\boldsymbol{\beta}}^T \left(\mathrm{Cov}(\boldsymbol{U_\beta}) - \mathrm{Cov}(\boldsymbol{U}_{\boldsymbol{\beta}}^\dagger)\right)\boldsymbol{t}_{\boldsymbol{\beta}} \end{aligned}$$

is computed only for those individuals with $g_i > 0$ which leads to a substantial reduction in running time from $O(n-m)$ to $O(m)$ calculations when $m << n$, which is typically the case, and particularly relevant for rare variants.

Similarly as for the normal approximation of $K(\boldsymbol{t}_{\boldsymbol{\beta}}, t_\gamma)$, the normal approximation of $K_{\boldsymbol{\beta}}(\boldsymbol{t}_{\boldsymbol{\beta}})$ is given by

$$K_{\boldsymbol{\beta}}(\boldsymbol{t}_{\boldsymbol{\beta}}) \approx \frac{1}{2}\boldsymbol{t}_{\boldsymbol{\beta}}^T \operatorname{Cov}(\boldsymbol{U}_{\boldsymbol{\beta}})\boldsymbol{t}_{\boldsymbol{\beta}}.$$

3 Simulations of genetic association studies with an imbalanced response

We will in detail explain the simulations given in Section 5.3. The simulation set-up is motivated by Dey et al. (2017) by conditioning on the which individuals are cases and controls, say $\boldsymbol{y} = (\boldsymbol{0}_c, \boldsymbol{0}_{n-c}^T)$, with $\boldsymbol{0}_c$ and $\boldsymbol{0}_{n-c}$ vectors of zeros with size c (the number of cases) and $n-c$ (the number of controls). We consider the logistic regression model

$$\operatorname{logit}(\mu_i) = \beta_0 + x_{i,1} + x_{i,2} + \gamma g_i,$$

with $X_1 \sim$ Bernoulli(0.5), $X_2 \sim N(0,1)$ and $G \sim \text{binom}(2, \text{MAF})$ mutually independent. Since we are evaluating validity of tests, we set $\gamma = 0$. For each iteration, the genotype vector is sampled independently of $\boldsymbol{y}$, while the nuisance covariates for each individual are sampled conditionally on $\boldsymbol{y}$. In each iteration, and for each method, we record whether a false rejection has occurred. With a total of 10^9 iterations, we estimate the type I error rate conditioned on the constant phenotype vector $\boldsymbol{y}$ when applying SPA, ESPA-CC and DSPA-CC. We want β_0 to be such that the disease prevalence to be 1% in the population, i.e. $P(Y=1) = 0.01$. That is we want:

$$\begin{aligned}
&P(Y=1) = \int_{x_1=-\infty}^{\infty} \sum_{x_1=0}^{1} P(Y=1|x_1,x_2)P(x_1,x_2)dx_1 \\
&= P(Y=1) = \int_{x_1=-\infty}^{\infty} \sum_{x_1=0}^{1} P(Y=1|x_1,x_2)P(x_1)P(x_2)dx_1 \qquad (5)\\
&= 0.5\sqrt{\frac{1}{2\pi}} \cdot \int_{x_2=-\infty}^{\infty} \exp(-0.5x_2^2)\left(\frac{1}{1+\exp(-\beta_0 - x_2)} + \frac{1}{1+\exp(-\beta_0 - 1 - x_2)}\right)dx_2 = 0.01
\end{aligned}$$

A solution is $\beta_0 = -5.6$.

Given the vector of phenotype values $\boldsymbol{y}$, the nuisance covariates need to be sampled according to their conditional probabilities:

$$\begin{aligned}
P(x_1|y) &= P(x_1,y)/P(y) = P(y|x_1)P(x_1)/P(y) \\
&= P(x_1)/P(y_1)\int_{x_2=-\infty}^{\infty} P(y|x_1,x_2)P(x_2)dx_2,
\end{aligned} \qquad (6)$$

and

$$\begin{aligned}
P(x_2|y) &= P(x_2,y)/P(y) = P(y|x_2)P(x_2)/P(y) \\
&= P(x_2)/P(y_1)\sum_{x_1=0}^{1} P(y|x_1,x_2)P(x_2),
\end{aligned} \qquad (7)$$

Therefore from (6) (with prevalence 0.01):

$$P(X_1 = x_1|y=1) = 50\sqrt{\frac{1}{2\pi}} \cdot \int_{x_2=-\infty}^{\infty} \frac{\exp(-0.5x^2)}{1+\exp(-\beta_0 - x_1 - x_2)}dx_2, \qquad (8)$$

and similarly one can compute $P(X_1 = x_1|y=0)$ for each value of x_1. For the sampling of x_2 conditional on y we will get for instance:

$$P(x_2|y=1) = 50\sqrt{\frac{1}{2\pi}}\exp(-0.5x^2)\left(\frac{1}{1+\exp(-\beta_0 - x_2)} + \frac{1}{1+\exp(-\beta_0 - 1 - x_2)}\right). \quad (9)$$

The continuous probability distribution of $P(x_2|y=1)$, as well as for $P(x_2|y=0)$, does not belong to any known distribution class, however one can see that $P(x_2|y=1) \leq \phi(x_2)$ for all x_2 with $\phi()$ the standard normal distribution. Therefore the standard normal can be used as a proposal distribution in a *rejection sampling* procedure. However, the large amount of simulations needed requires a faster approach as the efficiency in the rejection sampling depends on how close the proposal distribution resembles the true distribution. It can be shown in this particular case that the efficiency decreases for decreasing prevalence ($P(Y=1)$). Since the probability distribution of $P(x_2|y=1)$ as well as $P(x_2|y=0)$ can be shown to be log-concave, one can apply the much more efficient *adaptive rejection sampling* procedure, where the proposal distribution is adaptively improved during the iterations.

4 PCA plots

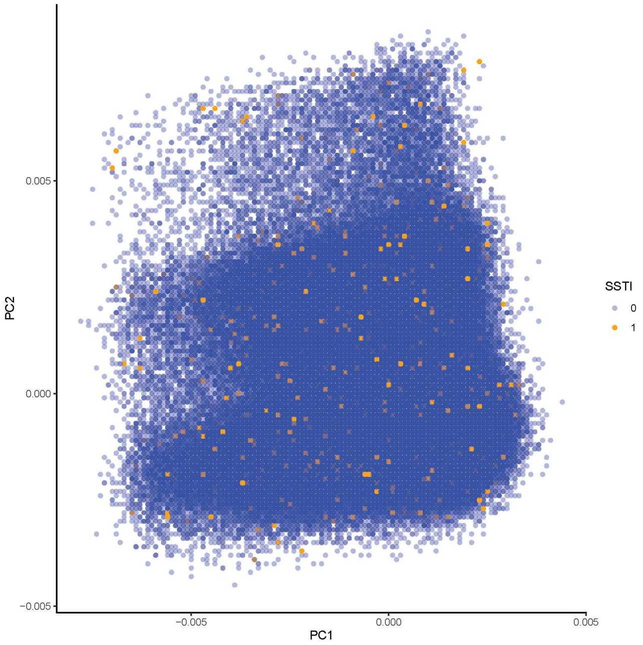


Figure 1: PCA plot of first and second principal components when analysing the common variants.

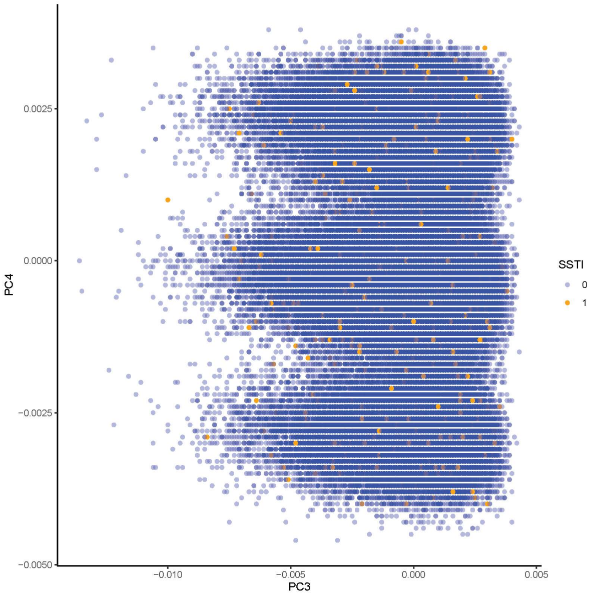


Figure 2: PCA plot of third and fourth principal components when analysing the common variants.

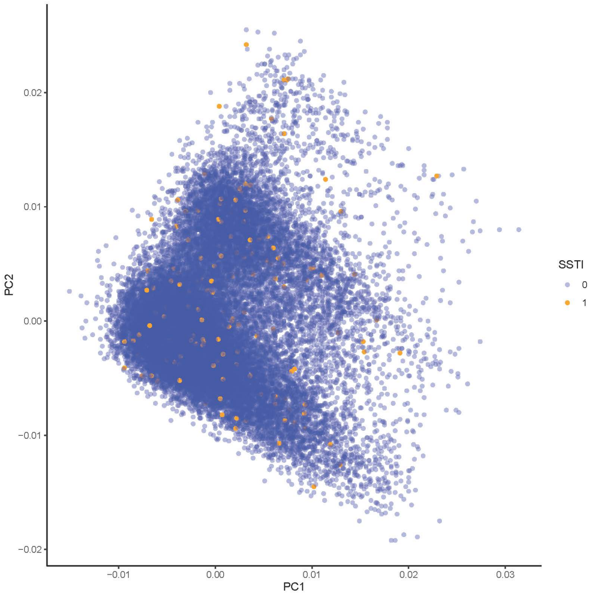


Figure 3: PCA plot of first and second principal components when analysing the rare variants.

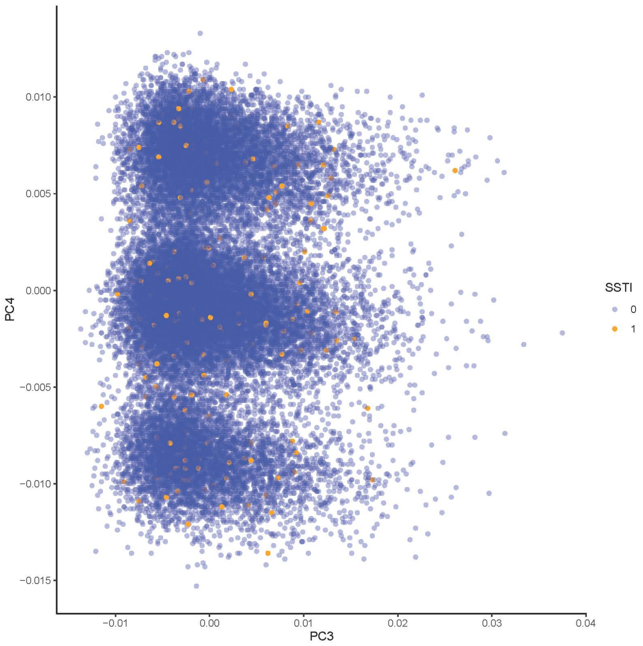


Figure 4: PCA plot of first and second principal components when analysing the rare variants.

References

Dey, R., Schmidt, E. M., Abecasis, G. R., & Lee, S. (2017). A Fast and Accurate Algorithm to Test for Binary Phenotypes and Its Application to PheWAS. *American Journal of Human Genetics*, *101*(1), 37–49.

Van Hout, C. V., Tachmazidou, I., Backman, J. D., Hoffman, J. X., Ye, B., Pandey, A. K., Gonzaga-Jauregui, C., Khalid, S., Liu, D., Banerjee, N., Li, A. H., Colm, O., Marcketta, A., Staples, J., Schurmann, C., Hawes, A., Maxwell, E., Barnard, L., Lopez, A., . . . on behalf of the Regeneron Genetics Center. (2019). *Whole exome sequencing and characterization of coding variation in 49,960 individuals in the UK Biobank* (tech. rep.).

Paper 2

A new method for exploring gene–gene and gene–environment interactions in GWAS with tree ensemble methods and SHAP values

Johnsen *et al. BMC Bioinformatics* (2021) 22:230
https://doi.org/10.1186/s12859-021-04041-7

BMC Bioinformatics

METHODOLOGY ARTICLE

Open Access

A new method for exploring gene–gene and gene–environment interactions in GWAS with tree ensemble methods and SHAP values

Pål V. Johnsen[1,2*], Signe Riemer-Sørensen[1], Andrew Thomas DeWan[3,4], Megan E. Cahill[3] and Mette Langaas[2]

*Correspondence: pal.johnsen@sintef.no
[1] SINTEF DIGITAL, Forskningsveien 1, 0373 Oslo, Norway
Full list of author information is available at the end of the article

Abstract

Background: The identification of gene–gene and gene–environment interactions in genome-wide association studies is challenging due to the unknown nature of the interactions and the overwhelmingly large number of possible combinations. Parametric regression models are suitable to look for prespecified interactions. Nonparametric models such as tree ensemble models, with the ability to detect any unspecified interaction, have previously been difficult to interpret. However, with the development of methods for model explainability, it is now possible to interpret tree ensemble models efficiently and with a strong theoretical basis.

Results: We propose a tree ensemble- and SHAP-based method for identifying as well as interpreting potential gene–gene and gene–environment interactions on large-scale biobank data. A set of independent cross-validation runs are used to implicitly investigate the whole genome. We apply and evaluate the method using data from the UK Biobank with obesity as the phenotype. The results are in line with previous research on obesity as we identify top SNPs previously associated with obesity. We further demonstrate how to interpret and visualize interaction candidates.

Conclusions: The new method identifies interaction candidates otherwise not detected with parametric regression models. However, further research is needed to evaluate the uncertainties of these candidates. The method can be applied to large-scale biobanks with high-dimensional data.

Keywords: GWAS, Tree ensemble models, XGBoost, SHAP, Model explainability, Gene–gene and gene–environment interactions

Background

In a traditional genome-wide association study (GWAS) each single nucleotide polymorphism (SNP) is tested individually for association with a particular phenotype. Using computationally efficient generalized or Bayesian linear mixed models that account for population stratification and cryptic relatedness, this approach can successfully identify risk alleles in the genome for complex diseases such as type 2 diabetes, Celiac disease and schizophrenia using large biobanks consisting of hundreds of thousands of individuals

© The Author(s), 2021. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/. The Creative Commons Public Domain Dedication waiver (http://creativecommons.org/publicdomain/zero/1.0/) applies to the data made available in this article, unless otherwise stated in a credit line to the data.

and SNPs [1–3]. Despite this, the estimated effects of the risk alleles are typically small and a large proportion of the estimated genetic heritability is yet to be explained for common traits and diseases [4]. One reason may be that most traits and diseases are highly polygenic, and thus many risk alleles with tiny effects will not be declared statistically significant due to stringent p-value significance thresholds. A second reason may be that the effect of the risk alleles are parametrically misspecified in the models. Model misspecification may lead to reduced power of detecting risk alleles [5, 6]. A third reason may be failure to account for *epistasis*, namely interactions between genes which together can impact the association with a certain phenotype [7, 8]. A fourth reason for the missing genetic heritability may be gene–environment interactions where the effect of a gene depends on some external environmental factor. Incorporating interactions in a generalized linear mixed model, particularly gene–gene interactions, remains a difficult task in GWAS due to the large number of interactions to investigate, the strict assumptions of the interaction effects needed and the multiple testing problem among other things [9].

In many situations the number of directly genotyped SNPs to evaluate, ignoring imputed genotype values, may be of the order of millions. With millions of SNPs to investigate the total number of SNP-pairs becomes of the order of 10^{12}. For instance, with a family-wise error rate (FWER) less that 0.05, using the Bonferroni method this will require rejection of the null hypothesis of no interaction for p-values less than 10^{-14}. Even with less conservative criteria than FWER, the small group of true interactions would be required to have very strong signals in order to be identified. Therefore, several two-stage algorithms have been developed such as GBOOST, SHEsisEpi and DSS where the first stage is a screening procedure to find the most promising gene–gene interactions, and the second stage is further investigation only based on these gene–gene interaction candidates [10–14]. However, inclusion of environmental features is either not considered or limited in the aforementioned two-stage algorithms [10, 15]. This can lead to overlooking important relationships including gene–environment interactions. Within modern biobanks, a rich amount of information, clinical, demographic, environmental and genetic, is available for each individual. A GWAS implemented using biobank data should therefore take full advantage of information with any perceived relevance for the trait of interest.

As an alternative to separately testing one parametric model for each interaction as well as the two-stage algorithms mentioned above, we suggest a nonparametric three-phase algorithm that can adjust for an unlimited number of features while searching for both gene–gene and gene–environment interaction candidates using tree ensemble models and SHAP values. We first rank the importance of each feature using the tree ensemble model XGBoost, a powerful prediction model suitable for high-dimensional data [16]. Recent research has demonstrated the possibility to interpret efficiently and with strong theoretical basis the importance of each feature from tree ensemble models using so-called SHapley Additive exPlanation (SHAP) values [17]. Based on this ranking, we further propose a model fitting process where the aim is to find the best XGBoost models with respect to predictive performance. The idea is that better predictive performance is a result of revealing additional relationships. Finally, based on these models, the aim is to explain the relationships that the models consider most important, and specifically the interactions. This type of procedure is more inclusive in order to find true interactions with the intention that these interaction candidates will need to be thoroughly investigated in a second stage. By using real data from UK Biobank, we demonstrate these models' capability to: (a) Rank features by importance and thereby removing noise. (b) Evaluate the use of XGBoost as both a predictive model and explainable model, and finally (c) Rank and explain plausible gene–gene and gene–environment interactions. We finish by comparing the top ranked interactions with logistic regression with interaction terms and perform statistical tests. We will in addition do a stratified analysis of the interaction candidates. In this paper, the focus is on a case-control setting, but the method outlined in this paper can be applied to both continuous and discrete phenotypes. Obesity was selected since this particular trait has been extensively researched in previous GWAS [18–20] providing a meaningful way to evaluate our method.

Methods

Recent research within GWAS to account for both genetic and environmental interactions have focused on how to explore the large amount of data in a more systematic way by using various nonparametric machine learning models such as tree ensemble models and deep neural networks [21–23]. So far, the most successfully applied machine learning methods for genotype data are tree ensemble models such as gradient tree boosting models [24] first introduced by Jerome H. Friedman [25], but with subsequent improvements. One such improvement is the so-called XGBoost implementation [16] used in this paper. XGBoost, as any tree ensemble model, consists of many so-called *weak learners* which in our case are *regression trees*. There are several advantages of using trees as they can naturally handle data of mixed type (continuous, categorical etc.) and missing values, they have the ability to deal with irrelevant and correlated variables, and they are computationally efficient to use [26]. However, trees suffer from low predictive power, high variance, lack of smoothness, and inability to capture linear structures. High variance and overfitting are of greater concern with deeper trees. Tree ensemble models, consisting of many trees, will reduce this variance and improve the predictive power [26]. Smoothness and ability to capture linear structures have also been shown to be improved [27]. The concern about using tree ensemble models within GWAS has been how to objectively evaluate the importance of each feature similar to p-values in traditional GWAS. However, a recent paper by Lundeberg et al. [17] showed that tree ensemble models have the capability to be efficiently and objectively interpreted by measuring the importance of each feature with respect to the predictions of the model by introducing so-called SHAP values. Interpretation of the XGBoost models through SHAP values will allow us to explain the prediction for each individual, a beneficial property in a precision medicine setting.

Problem description and syntax

Let y_i be the value/phenotype of some trait for individual i. This value may signify the absence or presence of a certain trait, such as a disease, or some continuous measure such as height, weight or blood pressure, or even a combination of measures such as the body mass index (BMI). Let $g_{i,a}$ denote the number of copies (0, 1 or 2) of the minor allele (referred to as the genotype) for a biallelic SNP a and individual i. Furthermore, let $x_{i,e}$ denote the value of some environmental feature, and let the matrix $\mathbf{X}_{N\times M}$ represent all genetic and environmental data for all N individuals and M features. Usually in a GWAS, the association between a SNP and a trait is tested separately for each SNP. However, another approach is to model the association between several SNPs and a trait simultaneously. We will use the latter approach, and will refer to genetic and environmental data as *features*, $\mathbf{x}_i$, for each individual i. Consider a model for predicting the phenotype, y_i, denoted $\hat{y}_i(\mathbf{x}_i)$. The performance of the model depends on how close each $\hat{y}_i(\mathbf{x}_i)$ is to y_i for all individuals with respect to some loss function. However, equally important in this setting is to understand what influences the prediction $\hat{y}_i(\mathbf{x}_i)$. In other words, we would like to understand how *each feature* contributes to the prediction $\hat{y}_i(\mathbf{x}_i)$ for each individual i. In this paper we aim to derive such a model and we will specifically consider the special case where the trait y_i is binary, that is, presence or absence of a phenotype. We denote the group consisting of individuals where the phenotype is absent as the *control group*, and the other group as the *case group*.

Before introducing our tree ensemble- and SHAP-based method for identifying interaction candidates, we will outline the necessary building blocks applied in our method including the choice of tree ensemble model, the performance metric to use in a binary classification setting as well as which metrics to use in order to evaluate the importance of each feature.

XGBoost

The XGBoost tree ensemble model consists of several regression trees, as illustrated in Fig. 1. An important aspect of trees, is that they automatically handle interactions between features. Consider the leftmost tree in Fig. 1, where the first split is for feature x_1, and then for both branches of the tree the next split is for feature x_2. Observe that the impact of feature x_2 in the tree is dependent on the value of feature x_1, with a different outcome if $x_1 \leq 1$ than if $x_1 = 2$. This means that a statistical interaction between feature x_1 and x_2 is encoded in the tree.

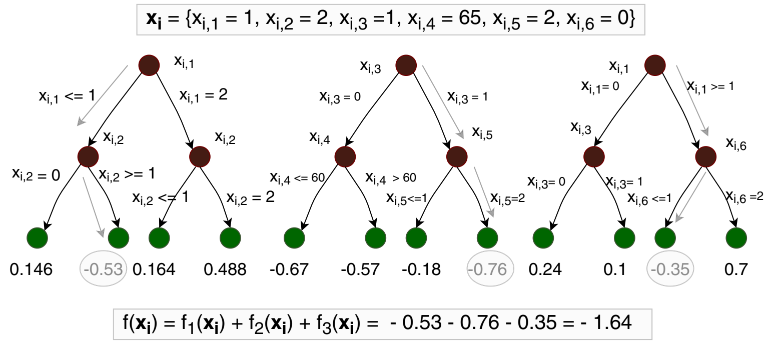


Fig. 1 An example with three constructed regression trees with six features $x_{i,1}$ to $x_{i,6}$ used as splitting points at each branch, and leaf node values. Also shown is the computation of $f(\mathbf{x}_i)$ given an example of feature values $\mathbf{x}_i$. The structure of the trees opens the possibility to explore interactions since a path from a root node to a leaf node denotes a combination of feature values

Constructing trees

The XGBoost algorithm starts with the construction of a single regression tree, and then new regression trees are consecutively constructed in a gradient boosting matter based on a loss function. The loss function is a sum of a loss function per individual, $\ell(y_i, \hat{y}_i^{(T)}(\mathbf{x}_i))$, which is a differentiable convex function. It measures the performance of the prediction, $\hat{y}_i^{(T)}(\mathbf{x}_i)$, with respect to the observed response, y_i, for observation i with features $\mathbf{x}_i$ when there is a total of T trees in the model. In a binary classification setting a convenient loss function is the binary cross-entropy:

$$\ell(y_i, \hat{y}_i^{(T)}(\mathbf{x}_i)) = -y_i \log(\hat{y}_i^{(T)}(\mathbf{x}_i)) - (1 - y_i) \log(1 - \hat{y}_i^{(T)}(\mathbf{x}_i)).$$

Regression tree number τ is denoted as f_τ, a data structure that contains information of nodes, features used as splitting points and leaf node values. The function $f_\tau(\mathbf{x}_i) \in \mathbb{R}$ outputs the value of the leaf node (green circles in Fig. 1) corresponding to features $\mathbf{x}_i$ based on tree τ. In a binary classification setting, the prediction $\hat{y}_i^{(T)}(\mathbf{x}_i)$ is interpreted as the probability that individual i is a case given a total of T regression trees.

In order for $\hat{y}_i^{(T)}(\mathbf{x}_i)$ to represent a probability, a much used transformation is the sigmoid function:

$$\hat{y}_i^{(T)}(\mathbf{x}_i) = \frac{1}{1 + e^{-\sum_{\tau=1}^{T} f_\tau(\mathbf{x}_i)}}. \tag{1}$$

When constructing each tree, one starts at the root node and successively investigates which feature to use as a splitting point at each node. The model will choose the split that minimizes the total loss function at that point. There are different strategies when constructing the trees. Splitting at the node which gives the largest decrease in loss is the approach that will be used in our case. The XGBoost R software package applies the histogram method to reduce the search time [28–30]. For the handling of missing values, we refer to the original XGBoost paper [16].

The model will typically stop training when the total loss function has not decreased in a given number of iterations, where a new regression tree is constructed in each iteration. The prediction of the final model on the logit scale given features $\mathbf{x}_i$ is given by $f(\mathbf{x}_i) = \sum_{\tau=1}^{T} f_\tau(\mathbf{x}_i)$, while the probability of the case class will be calculated using the sigmoid transform on $f(\mathbf{x}_i)$, as in Eq. (1).

Hyperparameters in XGBoost

XGBoost has a large set of hyperparameters, which may influence the performance of the algorithm and its ability to find the best representation of the data. In this paper, we focus on the learning rate η, *subsample*, *colsample_bytree*, *colsample_bylevel* and *max_depth*. The learning rate $\eta \in (0, 1]$ scales the values of the leaf nodes after the construction of each new tree, in which case $f_t(\mathbf{x}_i) = \eta f_t^*(\mathbf{x}_i)$ where $f_t^*(\mathbf{x}_i)$ is the raw regression tree before the scaling of the leaf node values has been applied. This will limit particular trees to dominate the prediction. It has been shown to be important since it influences how fast the model will learn and it can prevent early overfitting. In high-dimensional problems this is crucial and the learning rate should be well below 1 and is typically 0.1 or smaller [26, 31]. The subsample and colsample_bytree hyperparameters decide the proportion of individuals and features to be evaluated in each regression tree respectively. They also prevent overfitting, and in addition reduce the training time of the model. A typical value for both hyperparameters is 0.5, and in high-dimensional data it has been proposed that even smaller values can be beneficial [26]. However, this will depend on what proportion of the high-dimensional data is relevant. If the relevant proportion is small, a more reasonable value is closer to 1 [16]. The parameter colsample_bylevel is used to partition the number of possible features to use as splitting points in each level of the tree. The literature is quite scarce on its effect, but it may oppose the non-optimal greedy approach search as well as providing more room for learning in a way similar to the learning rate. The parameter max_depth is the maximum depth in each tree. Other important hyperparameters are the regularization parameter λ described in Chen and Guestrin [16] as well as the parameter early_stopping_rounds which is the maximum number of rounds without predictive improvement of the validation data before the training stops. To avoid overfitting, the validation data is independent of the training data.

Classification performance metric

For a binary classification model, the predictive performance in the validation data can be evaluated with specific focus on the group that is of particular interest (the case group). Let TP, FP and FN be the number of true positives, false positives and false negatives, respectively. The precision and recall given the classifications from a model are defined as follows,

$$\begin{aligned}\text{Precision } &= \frac{\text{TP}}{\text{TP} + \text{FP}},\\ \text{Recall } &= \frac{\text{TP}}{\text{TP} + \text{FN}}.\end{aligned}$$

A convenient measure for the model performance is the area under the curve, denoted PR-AUC (precision-recall area-under-curve) [32]. PR-AUC is most often used in the case of imbalance, meaning that one group is larger than the other. When TP $= 0$ and FP $= 0$, corresponding to a model that always predicts an individual to be a control, the precision is defined to be zero.

A measure of feature importance in tree ensemble models - SHapley Additive exPlanation (SHAP) values

When evaluating the global feature importance in a tree ensemble model, one possibility is to look at the relative decrease in loss for all splits by a given feature over all trees [33]. Unfortunately, this measure suffers from so-called *inconsistency* as discussed in Lundberg et al. [34]. In short, this means that the feature contributions are unfairly distributed as a result of not accounting for the importance of the order in which the features are introduced in the trees. Another popular, but similarly inconsistent, importance metric is counting the number of times each feature is used as a splitting point. Instead, a metric based on so-called SHapley Additive exPlanation (SHAP) values can be shown to achieve consistency [17, 35]. In the case of tree ensemble models, each feature j for each individual i is given a SHAP value, $\phi_{i,j}$, which represents the contribution of feature j with respect to the prediction, $f(\mathbf{x}_i) = \sum_{\tau=1}^{T} \eta f_\tau^*(\mathbf{x}_i)$, equal to the output of the linear sum of all T regression trees in a tree ensemble model given features $\mathbf{x}_i$. This metric exhibits several favourable properties aside from consistency [35]. For instance, the sum of the contributions of each feature, $\phi_{i,j}$, including a constant ϕ_0 equals the prediction of the model $f(\mathbf{x}_i)$:

$$f(\mathbf{x}_i) = \phi_0 + \sum_{j=1}^{M} \phi_{i,j}, \tag{2}$$

where M is the number of features included in the model. Moreover, the total contribution of a subset of all features for each individual is simply equal to the sum of the SHAP values for each feature. The reason for these favourable properties is that the contribution, $\phi_{i,j}$, is computed based on a concept from game theory first introduced by Lloyd Shapley [36]:

$$\phi_{i,j} = \sum_{\mathcal{S} \subseteq \mathcal{M} \setminus \{j\}} \frac{|\mathcal{S}|!(M - |\mathcal{S}| - 1)!}{M!} [v_i(\mathcal{S} \cup j) - v_i(\mathcal{S})], \tag{3}$$

where $\mathcal{M}$ is the set of all features included in the model, the function $v_i(\mathcal{S})$ measures the total contribution of a given set of features ($v_i(\mathcal{M}) = f(\mathbf{x}_i)$), and the sum is across all possible subsets where feature j is not included. The parameter ϕ_0 is defined as $\phi_0 = v(\mathcal{S} = \emptyset)$. The key idea is that the contribution of each feature for each individual is measured by evaluating the difference between the prediction when the value of feature j is known, versus the case when the value feature j is unknown for all subsets $\mathcal{S} \subseteq \mathcal{M} \setminus \{j\}$. In a statistical setting, the *marginal expectation* first introduced in Janzing, Minorics, and Blöbaum [37] seems to be a reasonable measure:

$$v_i(\mathcal{S} \cup j) = E[f(\mathbf{X}_{i,\mathcal{S}\cup\{j\}} = \mathbf{x}^*_{i,\mathcal{S}\cup\{j\}}, \mathbf{X}_{i,\overline{\mathcal{S}\cup\{j\}}})]$$

where $E[f(\mathbf{X}_{i,\mathcal{S}\cup\{j\}} = \mathbf{x}^*_{i,\mathcal{S}\cup\{j\}}, \mathbf{X}_{i,\overline{\mathcal{S}\cup\{j\}}})]$ is the expected prediction when only the values of the feature subset $\mathcal{S}$ as well as feature j, denoted $\mathbf{x}^*_{i,\mathcal{S}\cup\{j\}}$, are known, while the vector of the complement set, $\mathbf{X}_{i,\overline{\mathcal{S}\cup\{j\}}}$, is regarded as a random vector. Notice that $\mathcal{S} \cup \overline{\mathcal{S}} = \mathcal{M}$. The values $\phi_{i,j}$ in Expression (3) with $v_i(\mathcal{S})$ measured as marginal expectations are denoted as SHAP values [35]. In the case of binary classification using a tree ensemble model, the prediction $f(\mathbf{x}_i)$ can be interpreted as the log-odds prediction.

By assuming all features are mutually independent, Lundberg et al. [17] constructed an algorithm to estimate the SHAP values in polynomial running time, $O(TLD^2)$, with maximum depth D and maximum number of leaves L in all T trees. The assumption about mutual independence is a limitation, and without this assumption the estimation of the SHAP values becomes more complicated [38]. For further details about estimations of SHAP values assuming mutual independence, see Additional File 1.

SHAP interaction value

The SHAP values can be further generalized to interpret pairwise interactions through the SHAP interaction values $\Phi_{i,j,k}$, $j \neq k$, for individual i and features j and k given by [17, 39]:

$$\Phi_{i,j,k} = \sum_{\mathcal{S}\subseteq\mathcal{M}\setminus\{j,k\}} \frac{|\mathcal{S}|!(M-|\mathcal{S}|-2)!}{2(M-1)!}\nabla_{i,j,k}(\mathcal{S}), \tag{4}$$

where

$$\begin{aligned}\nabla_{i,j,k}(\mathcal{S}) = &\Big[E[f(\mathbf{X}_{i,\mathcal{S}\cup\{j,k\}} = \mathbf{x}^*_{i,\mathcal{S}\cup\{j,k\}}, \mathbf{X}_{i,\overline{\mathcal{S}\cup\{j,k\}}})]\\ &-E[f(\mathbf{X}_{i,\mathcal{S}\cup\{k\}} = \mathbf{x}^*_{i,\mathcal{S}\cup\{k\}}, \mathbf{X}_{i,\overline{\mathcal{S}\cup\{k\}}})]\Big]\\ &-\Big[E[f(\mathbf{X}_{i,\mathcal{S}\cup\{j\}} = \mathbf{x}^*_{i,\mathcal{S}\cup\{j\}}, \mathbf{X}_{i,\overline{\mathcal{S}\cup\{j\}}})] - E[f(\mathbf{X}_{i,\mathcal{S}} = \mathbf{x}^*_{i,\mathcal{S}}, \mathbf{X}_{i,\overline{\mathcal{S}}})]\Big].\end{aligned}$$

If feature k yields additional information when present simultaneously with feature j, $\nabla_{i,j,k}(\mathcal{S})$ will be different from zero with the sign depending on how feature k (when present) affects feature j. With these definitions, the pairwise SHAP interaction values have the same properties as the single-feature SHAP values. For instance, the contribution of a given feature j, $\phi_{i,j}$, can be separated into the contribution of j itself, denoted $\Phi_{i,j,j}$, in addition to all interactions including feature j, denoted as $\Phi_{i,j,k}$, for all $k \neq j$:

$$\phi_{i,j} = \Phi_{i,j,j} + \sum_{j\neq k} \Phi_{i,j,k}.$$

The final prediction for each individual can be decomposed into

$$f(\mathbf{x}_i) = \phi_0 + \sum_{j=1}^{M} \phi_{i,j} = \phi_0 + \sum_{j=1}^{M}\left[\Phi_{i,j,j} + \sum_{k\neq j}\Phi_{i,j,k}\right], \tag{5}$$

where $\Phi_{i,j,k} = \Phi_{i,k,j}$.

The interactions for all possible pairs of features for a particular tree ensemble model can be computed in $O(TMLD^2)$ time [17].

Tree ensemble- and SHAP-based method for identifying interaction candidates

We propose a new method using XGBoost and SHAP values to identify potential interactions such as SNP-SNP interactions or SNP-environment interactions, but also non-parametric single-SNP effects. The method is outlined in Fig. 2.

We use a tree ensemble model (XGBoost) trained on data consisting of observations from individuals each with a trait y_i and features $\mathbf{x}_i$, to rank features by importance using SHAP values. The ranked list of features makes it possible to construct new models that use only the most important features, and therefore have higher predictive power. Finally, having a fitted model that only consists of relevant features, we want to graphically present which relationships are important with respect to the phenotype, both marginal effects and interactions.

In order to evaluate the ability to both rank features by importance, find the best predictive models, and explain the best models without causing optimism bias, we divide the individuals in three disjoint subsets, namely the *ranking data*, *fitting data* and *evaluation data* (Fig. 3).

Dividing the data into several subsets will reduce the power to detect relevant features as well as reducing the degree to which each subset is representative of the

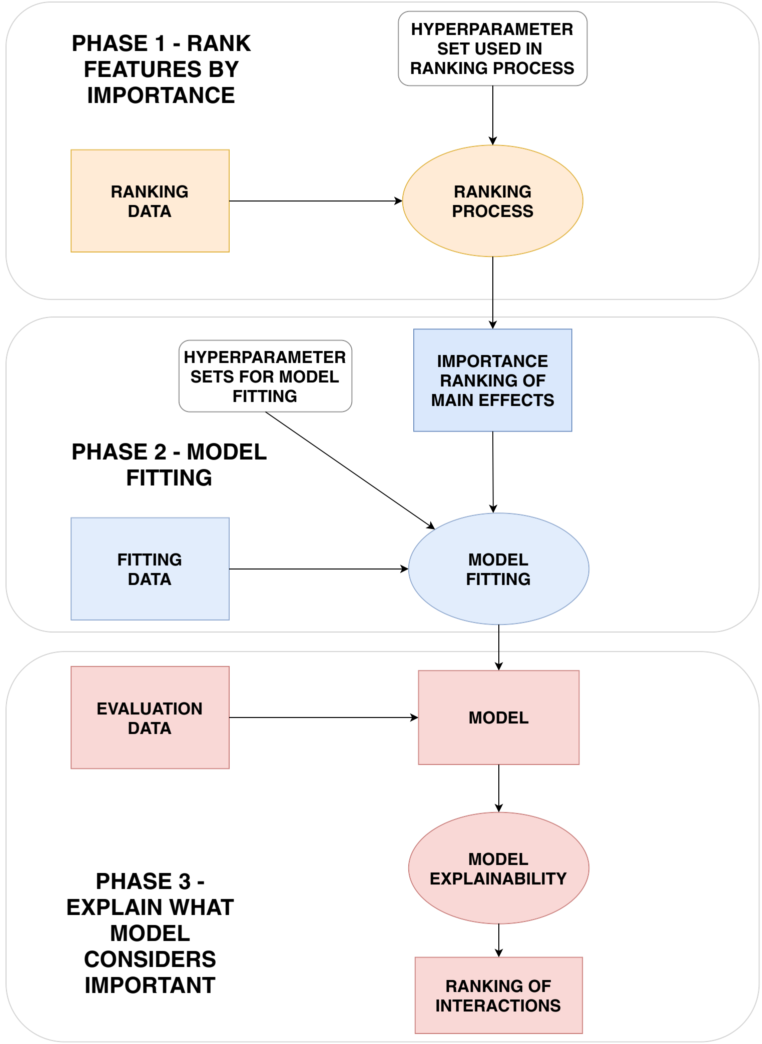

PHASE 1 - RANK FEATURES BY IMPORTANCE
HYPERPARAMETER SET USED IN RANKING PROCESS
RANKING DATA
RANKING PROCESS
HYPERPARAMETER SETS FOR MODEL FITTING
IMPORTANCE RANKING OF MAIN EFFECTS
PHASE 2 - MODEL FITTING
FITTING DATA
MODEL FITTING
EVALUATION DATA
MODEL
MODEL EXPLAINABILITY
PHASE 3 - EXPLAIN WHAT MODEL CONSIDERS IMPORTANT
RANKING OF INTERACTIONS

Fig. 2 The ranking, model fitting and explanation phases. In the ranking phase, the SNPs and environmental features are ordered by their relative importance. The ranking is achieved with XGBoost and SHAP values as explained in Fig. 4. In the model fitting process, the top ranked features are combined and modelled with XGBoost as described in Fig. 6. Finally, the explanations and interactions are obtained from the SHAP values. This is visualized in Figs. 10, and 11

full data set. However, the procedures are intended to be used on data from large biobanks to reduce power loss and representativeness of the subsets. By using independent subsets of the data for each phase of our method, we avoid potential overfitting by reusing data, and will be able to give an accurate account to which extent tree

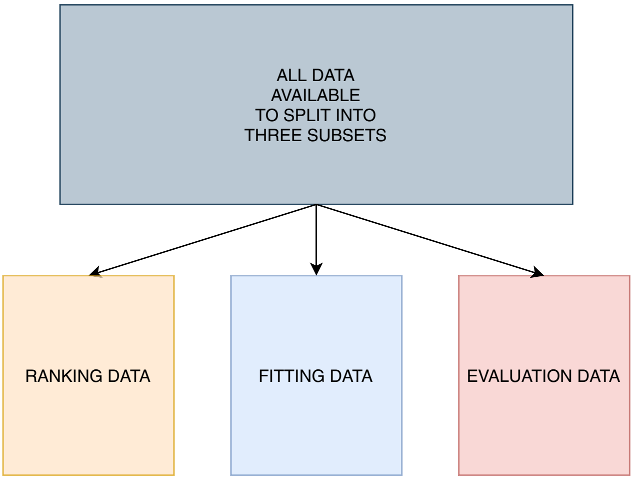


Fig. 3 All data available is divided into three subsets: Ranking data, fitting data and evaluation data. The ranking data is used to rank features by importance in order to remove noise. The fitting data is used to fit models by using the ranking derived from the ranking data. The evaluation data is finally used to explain what is considered important with respect to the predictions from the models trained on the fitting data

ensemble models are able to capture relationships between features and the trait of interest that classical GWAS methods might have difficulties to achieve [40].

Phase 1: The ranking process

Identifying associations between SNPs and a phenotype is a typical example of a high-dimensional problem. Experience from several GWAS suggests that many low-effect SNPs are not detected. At the same time we still expect only a small proportion of the total genome to have any effect with respect to the trait of interest. Consequently, we face a challenge where many potential SNPs have a causal effect on the trait, but a much larger number of SNPs are not causal at all and therefore contribute as noise. To make it even more complicated, among the large number of SNPs in the human genome, there exist correlations between different SNPs throughout the whole genome in a given population called *linkage disequilibrium* [41]. In general, the closer the physical distance between a pair of SNPs is, the more correlated the SNPs tend to be. As not all SNPs are genotyped, and if we disregard imputed data, there will be gaps between the SNPs that are present. We expect that in many cases, SNPs with causal effect fall in such gaps. But here we are helped by the linkage disequilibrium and the correlation between nearby SNPs. For practical purpose this means that a subset of all SNPs available can provide information beyond only those SNPs selected, but also those nearby SNPs that are in linkage disequilibrium. This also applies for interactions.

The analysis is further complicated by confounders such as population stratification and cryptic relatedness between individuals which can lead to spurious associations in our models [42]. Cross-validation is a model validation technique in which several models of identical structure are trained on different portions of the training data, and each model is evaluated on independent validation data. With respect to feature importance, a procedure with the purpose of preventing spurious associations, is to evaluate the importance of each feature based on all models constructed during cross-validation.

From our knowledge about linkage disequilibrium, population stratification and cryptic relatedness, we therefore propose a method to implicitly investigate the whole genome efficiently and objectively through a series of independent cross-validations by using XGBoost, a tree ensemble model, as shown in Fig. 4. It is from these independent cross-validations we will provide a ranking of the importance of each feature.

Consider a data set with N individuals and R directly genotyped SNPs. We create A randomly selected subsets, where each subset consist of S SNPs with low mutual correlation and $G \leq N$ individuals randomly sampled with equal probability in order to keep an as agnostic narrowed search as possible. Furthermore, each sampled subset is divided into F folds where $F - 1$ folds are used in an ordinary cross-validation to train $F - 1$ XGBoost models, while the last fold never seen or used during cross-validation is used as test data. This will create $F - 1$ models trained on different data, and their performance can be objectively evaluated on the test data. As shown in Fig. 5 for the $F - 1$ folds used in cross-validation, in each iteration $F - 2$ folds are used to train an XGBoost model, while the last fold is used as validation data. Training of the model will proceed as long as the performance on the validation

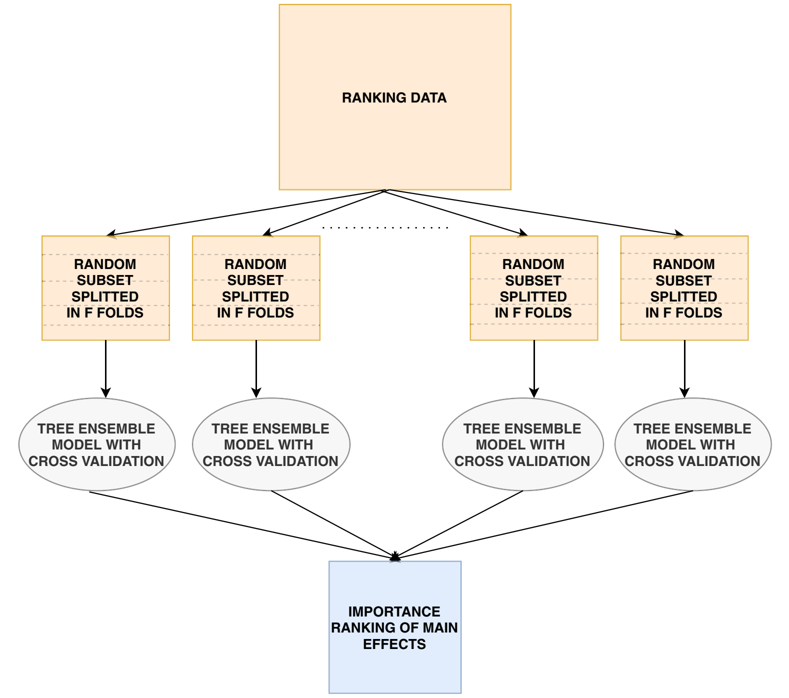

RANKING DATA
RANDOM SUBSET SPLITTED IN F FOLDS
RANDOM SUBSET SPLITTED IN F FOLDS
RANDOM SUBSET SPLITTED IN F FOLDS
RANDOM SUBSET SPLITTED IN F FOLDS
TREE ENSEMBLE MODEL WITH CROSS VALIDATION
TREE ENSEMBLE MODEL WITH CROSS VALIDATION
TREE ENSEMBLE MODEL WITH CROSS VALIDATION
TREE ENSEMBLE MODEL WITH CROSS VALIDATION
IMPORTANCE RANKING OF MAIN EFFECTS

Fig. 4 In the ranking process, multiple independent subsets are created and used in a cross-validation procedure with tree ensemble models. The trained models will be used to rank the importance of the features

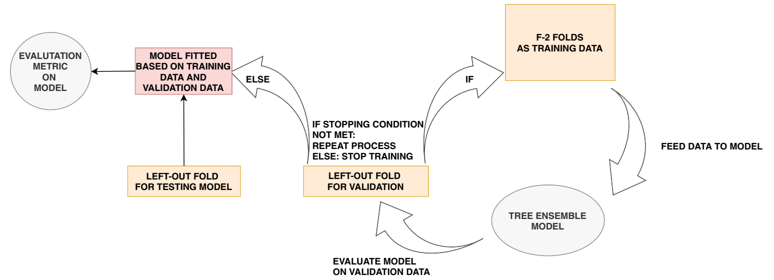

EVALUATION METRIC ON MODEL
MODEL FITTED BASED ON TRAINING DATA AND VALIDATION DATA
ELSE
IF
F-2 FOLDS AS TRAINING DATA
IF STOPPING CONDITION NOT MET: REPEAT PROCESS ELSE: STOP TRAINING
FEED DATA TO MODEL
LEFT-OUT FOLD FOR TESTING MODEL
LEFT-OUT FOLD FOR VALIDATION
TREE ENSEMBLE MODEL
EVALUATE MODEL ON VALIDATION DATA

Fig. 5 The cross-validation phase when training data consists of $F-2$ specific merged folds. Training of the model will proceed as long as the performance on the validation data improves within a certain number of iterations as given by the early_stopping_rounds hyperparameter

data improves within a certain number of iterations as given by the early_stopping_rounds hyperparameter. Cross-validation reduces the harm of both overfitting and selection bias [43]. The degree of overfitting can be further investigated by looking at the model performance difference on the validation and test data.

With A subsets each creating $F-1$ models, the question is now how to rank all features investigated in all A subsets for all $\mathcal{P} = A(F-1)$ models. We define a new concept called the *relative feature contribution*, denoted $\kappa_{i,j}^{p}$, for individual i, feature j and model p as:

$$\kappa_{i,j}^{p} = \frac{|\phi_{i,j}^{p}|}{|\phi_0^{p}| + \sum_{m=1}^{M} |\phi_{i,m}^{p}|}, \tag{6}$$

where $\phi_{i,j}^{p}$ is the SHAP value for feature j. The measure $\kappa_{i,j}^{p}$ can be interpreted as the proportion of the prediction for individual i attributed to feature j for model p. We now want to estimate the expected relative contribution of feature j using all the past independent cross-validations. The expected relative feature contribution (ERFC), $\hat{E}[\kappa_j]$, is defined as:

$$\hat{E}[\kappa_j] = \frac{1}{\sum_{p=1}^{\mathcal{P}} G_p I(j \in \sigma_p)} \sum_{p=1}^{\mathcal{P}} \sum_{i=1}^{G_p} I(j \in \sigma_p) \kappa_{i,j}^{p}, \tag{7}$$

where $\kappa_{i,j}^{p}$ denotes the relative feature contribution of feature j for individual i in a set of G_p individuals used to explain model p, and $I(j \in \sigma_p)$ is the indicator function which is equal to one if feature j is included in the subset data used to train model p, and zero elsewhere.

The individuals G_p used to explain a particular model p created from a particular subset a are chosen to be the individuals from the test data of the subset. This means that the contribution of each feature in each model will be based on individuals never seen during training. The estimation of $\hat{E}[\kappa_j]$ for each feature j will finally create a ranking of the contribution of each feature.

Phase 2: The model fitting process

Given a ranked list of features based on their feature contribution with respect to the trait of interest, this allows us to disregard irrelevant features and thus increases the ability to detect important relationships.

At this stage we are interested in finding the models with the best performance on some test data by utilizing the ranking of feature importance from the ranking process. For this purpose we use the fitting data never seen before in order to avoid any optimism bias [40]. The heterogeneity as well as possible relatedness among the individuals are taken into account by again using cross-validation. First we split the data in F folds, of which $F-1$ folds are used for cross-validation while the last fold is used as test data. This gives $F-1$ fitted models in total. The model fitting procedure is summarized in Fig. 6 which shows how one model (out of $F-1$) is fitted using only the top K features as well a set of hyperparameters. The aim is to find which set of $F-1$ models that on average performs best on the test data as a function of the value of K and hyperparameter values.

In order to explain the XGBoost models at a later stage we want to compute the SHAP values. We assume the features are mutually independent when computing the SHAP values. To take this into account, we combine the ranking with low values of the mutual squared Pearson's correlation, denoted r^2, when selecting the K features to include. See Sect. 2 in Additional File 1 for more information. Even though we are not guaranteed an independent set of features using r^2, it significantly limits the number of dependent features and therefore reduces the negative effect of misleading computations of SHAP values.

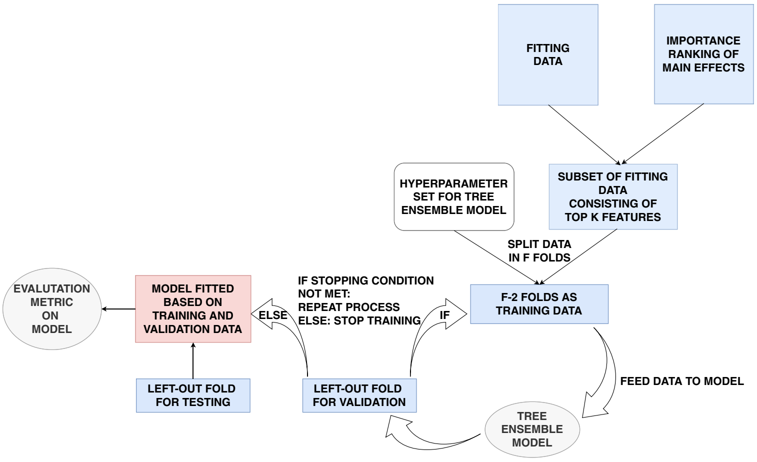

FITTING DATA
IMPORTANCE RANKING OF MAIN EFFECTS
HYPERPARAMETER SET FOR TREE ENSEMBLE MODEL
SUBSET OF FITTING DATA CONSISTING OF TOP K FEATURES
SPLIT DATA IN F FOLDS
EVALUTATION METRIC ON MODEL
MODEL FITTED BASED ON TRAINING AND VALIDATION DATA
IF STOPPING CONDITION NOT MET: REPEAT PROCESS ELSE: STOP TRAINING
ELSE
IF
F-2 FOLDS AS TRAINING DATA
FEED DATA TO MODEL
LEFT-OUT FOLD FOR TESTING
LEFT-OUT FOLD FOR VALIDATION
TREE ENSEMBLE MODEL

Fig. 6 Given a table of ranked feature importances, XGBoost models based on the top *K* features are trained in a new cross-validation procedure based on an independent set of individuals, namely the fitting data. We search for the XGBoost models that on average performs the best for a given set of hyperparameters (including the value of *K*) based on test data

Phase 3: Model explainability

After finding the best predictive models from the model fitting process, we can investigate which features and interactions contribute to the models through the SHAP values. Along the same lines as for the marginal feature importance used for ranking, the *relative contribution* for each interaction between feature j and k for a particular individual i and model p can be computed as:

$$\mu^{p}_{i,j,k} = \frac{2|\Phi^{p}_{i,j,k}|}{|\phi^{p}_{0}| + \sum_{m=1}^{M} |\phi^{p}_{i,m}|} \tag{8}$$

We can estimate the expected relative interaction contribution, $\hat{E}[\mu^{p}_{j,k}|G_e, p]$, given data consisting of G_e individuals and a model p:

$$\hat{E}[\mu^{p}_{j,k}|p, i = 1, ..., G_e] = \frac{1}{G_e} \sum_{i=1}^{G_e} \mu^{p}_{i,j,k}. \tag{9}$$

The G_e individuals are part of the evaluation data as shown in Fig. 2. As we have $F - 1$ models from the model fitting process, we average the result from all $F - 1$ models:

$$\hat{E}[\mu^{p}_{j,k}] = \frac{1}{F-1} \sum_{p=1}^{F-1} \hat{E}[\mu^{p}_{j,k}|p, i = 1, ..., G_e]. \tag{10}$$

We define this new concept as the expected relative interaction contribution (ERIC). This will provide a ranked list of interactions. A ranked list of marginal effects can be constructed as explained in the ranking process, but this time based on the $F - 1$ models constructed after the model fitting process.

The contribution of the top ranked marginal effects and interactions to the prediction for each individual can be visualized with sina plots and partial dependence plots as illustrated in Figs. 10 and 11 [17]. For one particular trained tree ensemble model, the sina plot in Fig. 10 shows the SHAP value for each individual indicated as a point with color depending on the value of the feature. The larger the absolute SHAP value, the more the feature contributes to the model prediction for a specific individual. Partial dependence plots, exemplified in Fig. 11, are used to visualize how the contribution, in other words the SHAP value, for a particular feature depends on another feature for different combinations of feature values. Here as well, each individual is marked as a point with the value of a given feature given on the x-axis and the corresponding SHAP value for this feature with respect to the prediction on the y-axis. The color of the point, however, represents the value of some other feature. In this way, interactions can be visualized and interpreted.

Results: application using UK Biobank data

As an example, we apply and evaluate the method described on data from the UK Biobank Resource [44]. Among the available phenotypes, obesity was chosen because it has been subjected to a number of high quality and well-powered GWAS that have identified more than 100 loci, many that have been consistently replicated across studies (e.g. FTO, BDNF, MC4R, TMEM18, SEC16B) [18–20] . Thus, we have a good set of true-positive loci with which to compare our results. We only analyzed White European individuals to limit the effect of population stratification. We define an individual to be part of either the control group ($y_i = 0$) or case group ($y_i = 1$) by:

$$y_i = \begin{cases} 1, \text{ if } 30 \leq \text{ BMI } \leq 70 \\ 0, \text{ if } 18.5 \leq \text{ BMI } \leq 25 \end{cases} \tag{11}$$

As should be evident above, we exclude overweight individuals with $25 <$ BMI < 30 from the analysis and only compare normal-weight individuals ($18.5 \leq$ BMI ≤ 25) with obese individuals (BMI ≥ 30). This reduces the number of subjects available for analysis, but allows us to define more distinct case and control groups. For power analyses of extreme phenotype data we refer the reader to [45]. The BMI data is provided from measurements at the initial assessment visit (2006–2010) at which participants were recruited and consent given. Phenotype-independent quality control of the genetic data for White European subjects consisting of the genotyped SNPs is completed using PLINK1.9 [46], and the details are given in Additional File 1. We only consider directly genotyped SNPs. In addition, we limit our analysis to SNPs with minor allele frequency (MAF) greater than 0.01. By only considering the two groups defined in Equation (11), this results in a total of 529 024 SNPs and 207 015 individuals to investigate, of which 43% of these individuals are in the group defined as obese. We apply the R package xgboost to both train xgboost models and to estimate SHAP values [47].

Environmental features

We include environmental features that are previously reported to be of importance with respect to obesity, namely sex, age, physical activity, intake of saturated fat, sleep duration, stress and alcohol consumption [48–52]. These environmental features are a representative set for the demonstration of the methodology and were not intended to be an exhaustive set of environmental features available in the UK Biobank for obesity. Information about the environmental features, including their definitions, are included in Additional File 1.

Ranking, fitting and evaluation data

We let the ranking data consist of 80,000 randomly chosen individuals, which will be used to rank the features by importance. The fitting data also consists of 80,000 individuals. This subset is used to find the best predictive models in the model fitting process. The evaluation data consists of 47 015 individuals, and is used to explain what the models found in the model fitting process consider the most important features and in which way they contribute. In all subsets, we retain the proportion of obese individuals.

Phase 1: The ranking process

By using the ranking data, at this stage we create $A = 50$ subsets where each subset consists of $G = 70{,}000$ individuals and $S = 110{,}000$ randomly chosen SNPs corresponding to 21% of the total number of SNPs available. The choice of total number of subsets to create is motivated from Eq. (2) in Additional File 1 with the criteria that any pair of SNPs appears in the same subset at least once with 90% certainty. The larger the number of individuals in each subset, the higher statistical power, but at the same time, the memory capacity limits the number of individuals in each subset at the cost of lost power. As the ranking process is time-consuming, we do not attempt any sophisticated hyperparameter optimization, but instead choose four hyperparameters sets that we regard as reasonable, given in Table 1. In addition, in all further analysis, the regularization parameter λ is set to 1, the default value in most XGBoost softwares [47]. The parameter early_stopping_rounds is set to 20.

As discussed in Blagus and Lusa [31], the learning rate η is set to be small for high-dimensional data such as 0.1, while as discussed in Chen and Guestrin [16], colsample_bytree is set to be large as there is only a small proportion of all features that are relevant. The hyperparameter subsample is also set to be large in order to increase the power to detect features of importance. The parameter colsample_bylevel has not been extensively discussed in the literature, but the parameter will oppose the greedy construction of the trees which may be beneficial in the long run. The maximum depth of the trees are set to no more than three, the reason being both computational considerations as well as the fact that the marginal expectations used to compute the SHAP values in (3) will be more inaccurate the deeper the trees are (see Additional File 1).

Using Eq. (7) to estimate the expected relative contribution for each feature, we give the ranking for the top 20 features in Table 2 for hyperparameter set 2 in Table 1.

Not surprisingly, the environmental features are considered most important. The next features are predominantly those connected to the FTO gene at chromosome 16 as expected from previous studies. A SNP close to the TMEM18 gene (rs13393304) is also found in the top 20 list. The next SNPs on the list are predominantly from chromosome 2, one SNP from chromosome 1 at the SEC16B gene (rs10913469) and further down SNPs from chromosome 18, yet no SNPs connected to the MC4R gene for instance. By further investigation, this is due to the fact that the SNPs randomly selected from the 50 subsets did not include any SNPs close to the MC4R gene which illuminates the issue when not creating enough subsets. Apart from this, one can see that the ranking process is able to detect small effects, and importance of each feature can be evaluated by computing SHAP values.

We compare with the corresponding ranked list derived using BOLT-LMM, a Bayesian mixed model that evaluates the marginal effect of each SNP, and computes p-values based on the BOLT-LMM infinitesimal mixed-model statistic [1]. The p-values are shown to be well-calibrated for significance levels as low as $5 \cdot 10^{-8}$ when the MAF of each SNP is larger than 1%, and that the case fraction is larger than 30% for a sample of 50,000 individuals [53]. All these criteria are satisfied in our ranking data set (with case fraction 42%, MAF greater than 1% and 80,000 individuals). Table 3 shows the top ranked 13 SNPs (top environmental features are not listed) where features with the smallest p-values are regarded to be of most importance.

In this case, all SNPs are related to the FTO gene, and most of the SNPs except two are also present in Table 2. These two SNPs were not sampled in any subset from the ranking process. The ordering in Table 2 and 3 between SNPs related to the FTO gene are slightly different. However, at this stage it is not strictly necessary to find the true order of the feature impacts, but an approximate order that allows us to discard features with insignificant impact in the further analysis.

Evaluation of the trained models used in the ranking process

To explore the degree of overfitting of the models trained during the ranking process, the PR-AUC score of each model computed on its corresponding validation data and test data (see Fig. 5) are explored in a Bland–Altman (mean—difference) plot. This shows the average PR-AUC score for each model on the x-axis, and the difference between the two scores on the y-axis. The results for all chosen sets of hyperparameters given in Table 1 can be seen in Fig. 7.

Figure 7 shows no clear pattern of overfitting as can be seen from the agreement between the density plots of the difference in PR-AUC scores. However, hyperparameter set 1 from Table 1 shows a cluster of bad predictions with PR-AUC around 0.56. The reason for this can be seen in Fig. 8 where bad predictions using hyperparameter set 1 is due to early stopping in the training. When there is no early stopping in the training, we also see that due to the small learning rate given in set 1, more trees are constructed than for the other hyperparameter sets, but yet the performance score is not superior. This emphasizes the importance of hyperparameters.

Phase 2: model fit from the ranking process and from BOLT-LMM ranking

In the model fitting process, we use the fitting data to train new XGBoost models with cross-validation by including the K most important SNPs for $K = 0$ (only including environmental features), $K = 100, 500, 1000, 3000, 5000, 10{,}000$ and finally $K = 15{,}000$. The ranking of the features is the output of the ranking process. In addition, to assess the quality of our method, we also train models based on the ranked table produced by BOLT-LMM.

Before training, the set of the K chosen SNPs is reduced such that the SNPs have mutually squared Pearson's correlation $r^2 < 0.2$ (see Additional File 1 for practical details about implementation). Due to computational limitations, we will only consider

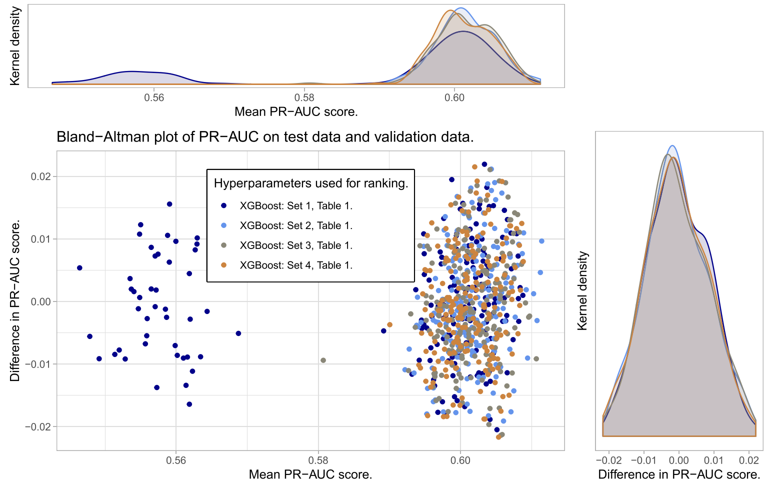


Fig. 7 Bland–Altman plot for the trained models used for ranking. No clear signs of overfitting, but one set of hyperparameters shows one cluster of poorer predictions than the others

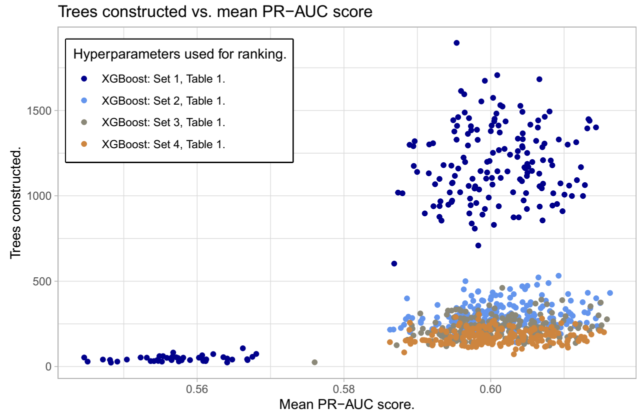


Fig. 8 The reason some models with hyperparameter set 1 in the ranking process underperforms is early stopping of the training. Notice also that a larger number of trees need to be constructed to get the same performance as for models with other hyperparameter sets

hyperparameter tuning from the XGBoost models through the sets given in Table 4, and optimize based on these sets. For each K and for the ranking based on our method and the ranking based on the BOLT-LMM model, the maximum average PR-AUC-score for the XGBoost models constructed in the cross-validation is found among the possible hyperparameter sets. For each K, we compare how the predictive model perform on the held-out test data from the fitting data. The results are shown in Fig. 9. When we vary K from small to large values, we expect that the model performance increases the most at the beginning as the most influential features are included, while as more features with low importance are added, the performance increases steadily until it flattens. At the end, the performance may even decrease as noise are added to the model in the form of SNPs without any predictive power.

The turning point for the BOLT-LMM ranking is $K = 1000$ while for the models based on the ranking process the turning point is consistently for a larger K value. The maximum average PR-AUC-score for the XGBoost models created in cross-validation is in general larger when using the ranking based on our method than the ranking based on BOLT-LMM. From Fig. 9, the average performance score is in general better when allowing the regression trees to be of maximum depth three instead of two. Additionally, inclusion of the SNPs provide only a small contribution to the increase in the average prediction performance, where the best models increase the average PR-AUC score from 0.606 when only environmental features are included to 0.629 when the top 5000 SNPs are included (blue line, Fig. 9b). This corresponds to an increase in average classification accuracy from 0.64 to 0.66.

Phase 3: Model explainability

In the model explainability phase we use the evaluation data consisting of 47,015 individuals, that has not been used in Phase 1 and 2. For convenience, we consider the models constructed during cross-validation that performed best on average on the test

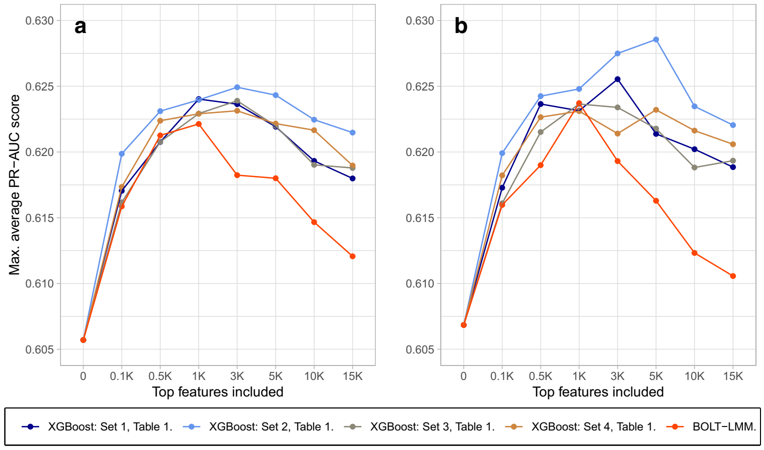


Fig. 9 The model fitting process based on top *K* features from both the ranking process (for different sets of XGBoost-hyperparameters indicated by the different colours and the legend) and from BOLT-LMM, for different values of *K*. In **a** hyperparameter sets 1–8 (all with max_depth = 2) from Table 4 in the model fitting process are used. In **b** hyperparameter sets 9–16 (max_depth = 3) are used. Both figures show that the use of the ranking process gives in general better model performance than for the BOLT-LMM ranking. There is also some gain in performance by increasing the hyperparameter max_depth from two to three

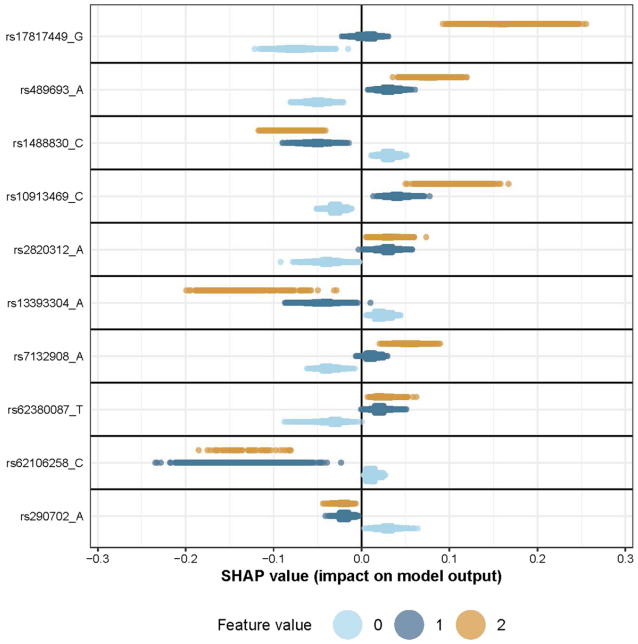


Fig. 10 A sina plot visualise the importance of each feature from a fitted model. Here we show the sina plot of the marginal effects for one of the four models constructed during the model fitting process when applied to the evaluation data from UK Biobank

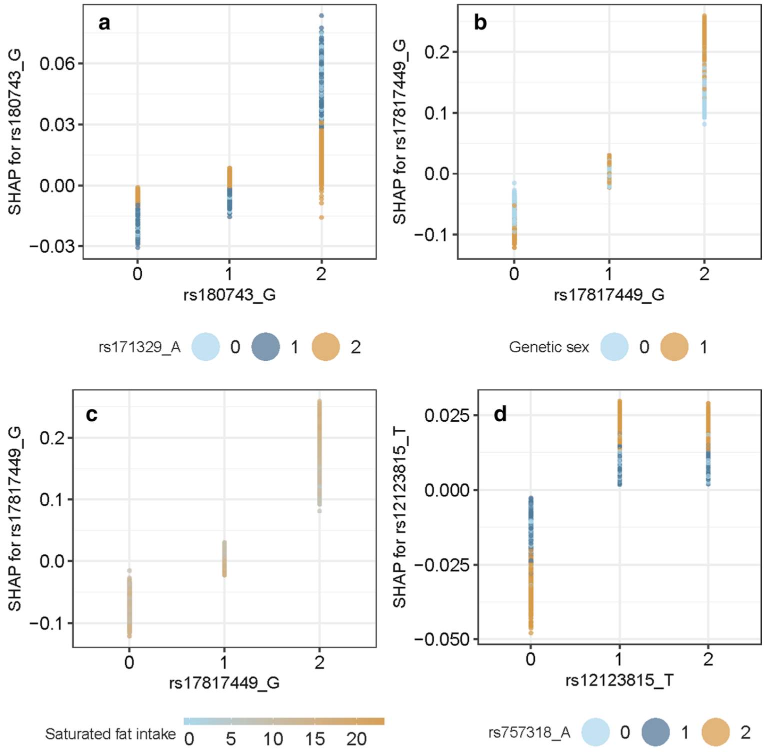


Fig. 11 Partial dependence plots for the pairs **a** rs180743 and rs171329, **b** rs17817449 and genetic sex, **c** rs17817449 and saturated fat intake, and **d** rs12123815 and rs12123815. In all panels we see how the SHAP values (vertical axis) depends on the feature value of the SNP (horizontal axis) and on the value of the second feature (color)

data during the model fitting process. These are the four models from fourfold cross-validation trained on the top 5000 ranked features with hyperparameter set 2 visualised as the blue line in Fig. 9b. We now explore what these four models consider important with respect to their predictions on the evaluation data. This is done by computing the expected relative contribution for both individual features as well as interactions. Marginal and interaction effects can be visualized with sina plots and partial dependence plots respectively. For the case of marginal effects, Fig. 10 shows the sina plot for one of the four models trained on the SNPs with the largest expected relative contributions. Here, we visualize both dominant and additive main effects found by our nonparametric method.

We use Eq. (8) together with Eq. (10) to compute the average relative interaction contribution (ERIC) for each pair of features based on the evaluation data, and list the top 10 interaction candidates in Table 5.

First of all, we see that the contributions from the interactions are quite small with expected relative interaction contribution (ERIC) of no more than 0.001. To further investigate the behaviour of these interaction candidates, in Fig. 11 we show partial dependence plots [17, 26] for the top four interactions from Table 5 when regarding one specific chosen model, out of the four, for each interaction.

We see in Fig. 11 examples where the SHAP value of the feature for each individual represented along the x-axis not only depends on its own feature value, but the value of some other feature as well. For instance, in Fig. 11a, we see that the increased risk of being obese when the genotype value is equal to two for rs180743, is reduced if the genotype value of rs171329 is equal to two as well. We also see in Fig. 11b that being a male (orange points) gives higher risk of being obese when the genotype value of rs17817449 is two, compared to when the genotype value is zero or one. A positive SHAP value implies a positive contribution to the log-odds prediction, and therefore a contribution making it more likely to be a case (obese).

Interaction models in logistic regression

We compare the interaction rankings from Phase 3 with logistic regression fits on the full UK Biobank data set and the evaluation data alone. We consider a parametric model, assuming additive effects, for both SNP-SNP and SNP-environment interaction effects for logistic regression, and construct a hypothesis test to infer the presence of interactions. For the test of SNP-SNP interactions between two SNPs a and b, the null model will be:

$$\text{logit}_{H_0,add}(P(Y_i = 1|g_{i,a}, g_{i,b}, \mathbf{x}_{i,c})) = \mathbf{x}_{i,c}^T\gamma + \alpha g_{i,a} + \beta g_{i,b}, \tag{12}$$

where $\mathbf{x}_{i,c}^T$ is a vector of features such as intercept, age, environmental features and principal components, while γ is the vector of corresponding parameters for each feature. The parameters α and β are the marginal effects from SNP a and b resepectively. The corresponding alternative model incorporating an additive interaction effect will be:

$$\text{logit}_{H_1,add}(P(Y_i = 1|g_{i,a}, g_{i,b}, \mathbf{x}_{i,c})) = \mathbf{x}_{i,c}^T\gamma + \alpha g_{i,a} + \beta g_{i,b} + \nu g_{i,a} g_{i,b}. \tag{13}$$

For a SNP-environment interaction we will use the following alternative model:

$$\text{logit}_{H_1}(P(Y_i = 1|g_i, x_{i,e}, \mathbf{x}_{i,c})) = \mathbf{x}_{i,c}^T\gamma + \alpha g_{i,a} + \beta_e x_{i,e} + \phi g_{i,a} x_{i,e}, \tag{14}$$

where β_e and ϕ are marginal environmental effect and interactions parameters respectively.

For the testing of the interactions we apply the likelihood ratio test (LRT) to test the null hypothesis that $\nu = 0$ for SNP-SNP interactions or $\phi = 0$ for SNP–environment interactions [26, 54]. The LRT assumes independence between the samples, and so we need to make sure the individuals included in the test are not related to any significant degree.

Comparison of Phase 3 results with logistic regression tests

Let the vector $\mathbf{x}_{i,c}$ given in (13) consist of the intercept in addition to the features sex, age and the top four principal components for each individual. The principal components are used to correct for population stratification [55]. The ranking of the pairwise interactions is based on the evaluation data consisting of 47,015 individuals. We fit a logistic regression model based on all unrelated individuals in the evaluation data (39,286 individuals), as well as a logistic regression based on all unrelated individuals used in this paper (173,468 individuals). Unrelatedness is ensured by using data field 22020 in the UK Biobank Data Showcase [44]. The principal components were calculated using EIGENSOFT (version 6.1.4) SmartPCA [56, 57]. We compute the principal components on the unrelated individuals in the evaluation data and all unrelated individuals separately. PCA plots for both the evaluation data and the full data set can be seen in the Additional File 1. A few individuals have missing values for each test and are removed.

The top four interactions from the SHAP values visualized in Fig. 11 are evaluated by applying likelihood ratio tests for each interaction. The results are given in Table 6.

It is clear that the sample size is the dominating factor for the computed *p*-values. All *p*-values based on the evaluation data, the same data that is used to rank the interactions, are non-significant. As expected, the *p*-values are in general smaller when considering all individuals, yet none of them would be declared significant in the case of any reasonable genome-wide multiple testing procedure [58]. The smallest *p*-value is achieved for the interaction between the SNP rs17817449 and genetic sex when including all individuals. In the Additional File 1, we apply likelihood ratio tests based on logistic models with less stricter assumptions, but with the need for more parameters. However, this does not provide smaller *p*-values to any significant degree. The reason may be that these tests are less powerful due to a higher number of degrees of freedom [54].

Stratified analysis

Instead of incorporating prespecified interactions in the logistic regression model, one can instead stratify in groups according to the value of a feature a, and investigate the effect of a feature b for each group. For instance, one can fit for each group a logistic regression model with respect to feature b such as in (12). For a true interaction, the log odds ratio of feature b will differ between some or all groups. Fig. 12 shows a stratified analysis for the top four interactions in Table 5, with 95% confidence intervals assuming normality of the estimated log odds ratios, adjusting for the same environmental features. The first example where the log odds ratio of rs171329 is compared within stratified groups of rs180743 do not change additively, the opposite of what is assumed in (13). However, the second example concerning rs17817449 and sex do show additive changes in the log odds ratios. The third interaction also shows small, yet indicative, differences in the log odds ratios. In the last example with rs12123815 and rs757318 the uncertainties in the log odds ratios are too large to give any conclusion.

Discussion

We have proposed how tree ensemble models, such as XGBoost, can be combined with SHAP values to explain the importance of individual SNPs as well as gene–gene and gene–environment interactions. The method has been illustrated on an example from the UK Biobank. We have shown that through several independent cross-validations on XGBoost models using subsets of SNPs spread along the genome, one is able to find a reasonable ranking of individual SNPs similar to what is found in previous GWAS of obesity [18]. In fact, Fig. 9 suggests that the ranking process has the potential to outperform BOLT-LMM.

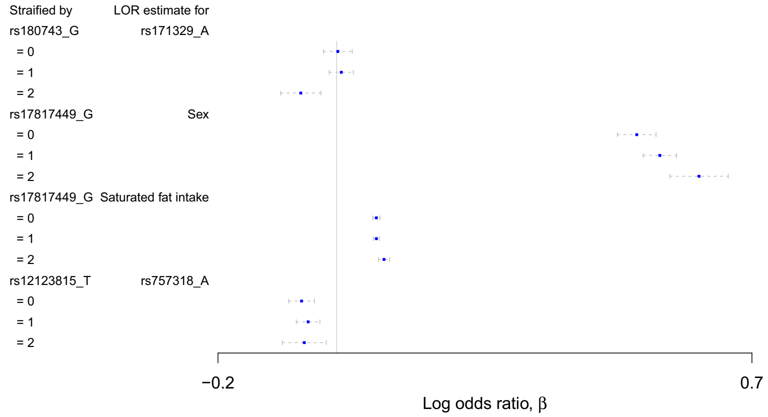


Fig. 12 Stratified analysis of the top four interactions based on all unrelated individuals to illustrate how the log odds ratio, with 95% confidence intervals, of one feature changes depending on the value of another feature

Table 1 The four hyperparameter sets for XGBoost considered in the analysis during the ranking process

Set	η	colsample_bytree	subsample	colsample_bylevel	max_depth
1	0.01	0.9	0.9	0.9	2
2	0.05	0.8	0.8	0.8	2
3	0.05	0.8	0.8	0.8	3
4	0.1	0.8	0.8	0.8	2

Ranking of interactions through SHAP values

The SHAP values may also identify interactions, but further investigation is needed. Comparing the top ranked interactions with logistic regression including interaction parameters, we see that none of the corresponding statistical tests provide convincing *p*-values. Assuming the ranking of interactions via SHAP values is reliable, we see from Table 5 that the interaction effects are small. Any genome-wide multiple testing procedure would struggle to find such small interaction effects. In addition, misspecification of the effects in the logistic regression models may reduce statistical power. Figure 12 shows that only the potential interaction between rs17817449 and sex seems to be additive. Tree ensemble models do not have any presumptions of what kind of effects are present, but instead they learn the effects iteratively. These effects can be investigated efficiently through SHAP values. However, the SHAP values are estimated, and uncertainties in these estimates must be accounted for. The interaction between rs12123815 and rs757318 in Fig. 12 is an example that may very well be a false positive. There is therefore a need to develop tests that can infer the trustworthiness of the SHAP values in a similar fashion as through *p* values. The development of such tests will be important future research within SHAP values.

Table 2 The resulting ranking based on the expected relative feature contribution (ERFC) from the ranking process for hyperparameter set 2 in Table 1

Feature	ERFC
Sex	0.12
Alcohol intake frequency	0.12
Physical activity	0.11
Saturated fat intake	0.058
Stressful events	0.056
Sleep duration	0.049
Age at initial assessment	0.047
rs17817449 (FTO, Chr. 16)	0.025
rs1421085 (FTO, Chr. 16)	0.025
rs1121980 (FTO, Chr. 16)	0.024
rs7202116 (FTO)	0.023
rs9941349 (FTO)	0.023
rs9940128 (FTO)	0.023
rs9922619 (FTO)	0.023
rs13393304 (FAM150B - TMEM18, Chr. 2)	0.022
rs12149832 (FTO)	0.021
rs9939609 (FTO)	0.021
rs9930506 (FTO)	0.021
rs11642841 (FTO)	0.020
rs2947411 (Chr. 2)	0.019

The environmental features are, as expected, considered more important than the SNPs, while the most important SNPs are at or nearby the FTO gene in agreement with previous studies

Table 3 The result after running BOLT-LMM on the ranking data showing the top SNPs with smallest *p*-value from the BOLT-LMM infinitesimal mixed-model statistic

Feature	BOLT-LMM *p*-value
rs1421085 (FTO)	3.7E−57
rs9940128 (FTO)	1.8E−54
rs1121980 (FTO)	2.4E−54
rs3751812 (FTO)	7.0E−54
rs17817449 (FTO)	8.5E−54
rs9939609 (FTO)	1.3E−53
rs8050136 (FTO)	2.2E−53
rs7202116 (FTO)	5.7E−53
rs9941349 (FTO)	5.0E−52
rs12149832 (FTO)	3.0E−50
rs9922619 (FTO)	1.0E−48
rs9930506 (FTO)	1.1E−48
rs11642841 (FTO)	1.3E−40

All top SNPs are connected to the FTO gene

One natural way to account for some of the uncertainties in the SHAP values is through cross-validation. In addition, larger absolute SHAP values may not only be as a consequence of larger feature importance, but also as a consequence of larger uncertainties in the SHAP values. The denominator in ERFC and ERIC, given in (7)

Table 4 The hyperparameter sets considered during the model fitting process

Set	η	colsample_bytree	Subsample	colsample_bylevel	max_depth
1	0.1	0.3	0.3	0.3	2
2	0.1	0.5	0.5	0.5	2
3	0.1	0.5	0.5	1	2
4	0.1	0.8	0.8	0.8	2
5	0.1	1	1	1	2
6	0.05	0.5	0.5	0.5	2
7	0.05	0.8	0.8	0.8	2
8	0.2	0.5	0.5	0.5	2
9	0.1	0.3	0.3	0.3	3
10	0.1	0.5	0.5	0.5	3
11	0.1	0.5	0.5	1	3
12	0.1	0.8	0.8	0.8	3
13	0.1	1	1	1	3
14	0.05	0.5	0.5	0.5	3
15	0.05	0.8	0.8	0.8	3
16	0.2	0.5	0.5	0.5	3

Table 5 The top 10 interactions based on the expected relative interaction contribution (ERIC) estimated on the evaluation data (Phase 3), with the aim of explaining the best predictive models from Phase 2

Feature 1	Feature 2	ERIC
rs171329	rs180743	0.001
Sex	rs17817449	0.001
Saturated fat intake	rs17817449	0.00094
rs757318	rs12123815	0.0008
rs4697952	rs1488830	0.00074
rs60822591	rs17854357	0.00066
rs4711329	rs11676272	0.00066
rs1518278	rs1488830	0.0006
Sex	rs12123815	0.00056
rs7132908	rs9949796	0.00054

Table 6 Results from likelihood ratio tests applied on the top four ranked interactions found from the model explainability process based on the evaluation data

Data set	Interaction	*p*-value LRT
Evaluation data	rs171329 and rs180743	0.85
All individuals	rs171329 and rs180743	0.024
Evaluation data	rs17817449 and genetic sex	0.77
All individuals	rs17817449 and genetic sex	4.09e-05
Evaluation data	rs17817449 and saturated fat intake	0.44
All individuals	rs17817449 and saturated fat intake	0.0017
Evaluation data	rs757318 and rs12123815	0.25
All individuals	rs757318 and rs12123815	0.71

and (10), equal to the sum of the absolute SHAP values for each individual will tend to be larger, the larger the variance of the SHAP value estimates are. Consequently, the importance measures ERFC and ERIC are reduced for increasing uncertainties in the SHAP values.

Data split

In this paper, data is split in three subsets used for ranking, model fitting and model explanation respectively. This procedure requires a large amount of data, but the purpose was to evaluate the credibility and potential of using tree ensemble models together with SHAP values. For smaller data samples, an alternative procedure is to rank interactions directly during the ranking process by computing the expected relative interactions contributions (ERIC). However, the ranking process consists of many models with low predictive power, which makes it more difficult to explore the true relationships compared to the models constructed in the model fitting process.

Limitations and improvements

The choice of number of SNPs *S*, individuals *G*, folds *F* and r^2-threshold in each cross-validation in the ranking process are all important with respect to performance, and should be considered as hyperparameters. The number of SNPs *S* must be large enough to represent important regions in the genome, but not so large that it introduces noise to the model. The number of individuals in each cross-validation, *G*, should be as large as possible as it increases the power to detect small as well as nonlinear effects. However, that may lead to computational challenges. The number of folds in the cross-validations, *F*, should neither be too small nor too large as we want to train the model on as many different subsets of the population as possible in order to find the most general effects, but at the same time the validation data set must be large enough to be sufficiently representative.

The mutual independence assumption when computing the SHAP values is a significant restriction, and a mutual r^2 below any threshold between features will by no means ascertain mutual independence as r^2 measures linear dependency. Correlation measures that can also account for nonlinear dependencies in a high-dimensional setting could provide more trustworthy results.

Hyperparameter optimization

We have seen that the hyperparameters for XGBoost are important. Unfortunately, the computation time for each set of hyperparameters is protracted, and consequently systematic hyperparameter optimization is not feasible. However, from the choice of hyperparameter sets in this paper, the hyperparameters colsample_bytree, subsample and colsample_bylevel should be high (0.8–0.9), while the learning rate η should be low (0.05–0.1), but not too low. Another important hyperparameter, the regularization parameter, λ, should be investigated more extensively.

Predictive performance and obesity

Even with strong predictors such as physical activity, intake of saturated fat, alcohol use, stressful events, sleep duration, age and sex in addition to genome-wide genetic data, we are not capable of constructing a model with more than 66% classification accuracy, and the genetic data only provide a small portion of the predictive performance. The usefulness lies in the fact that tree ensemble models can be used to identify nonparametric gene–gene and gene–environment interaction candidates while accounting for a large amount of features simultaneously. If the prediction performance of the model is considered satisfactory, this can be an important diagnostic tool in the future.

Conclusion

Our proposed tree ensemble- and SHAP-based method gives us the possibility of exploring both gene–gene and gene–environment interactions without any presumptions of what kind of effects may be present as well as adjusting for environmental features. Our proposed method can be applied to high-dimensional genetic data in large-scale biobanks. There is however a need to develop methods for assessing the uncertainties of the SHAP values to conclude whether the interaction candidates are reliable.

Abbreviations

BMI: Body mass index; ERFC: Expected relative feature contribution; ERIC: Expected relative interaction contribution; FN: False negative; FP: False positive; FWER: Family-wise error rate; GWAS: Genome-wide association study; LRT: Likelihood ratio test; MAF: Minor allele frequency; PCA: Principal component analysis; PR-AUC: Precision-recall area-under-curve; SHAP: Shapley additive explanation; SNP: Single nucleotide polymorphism; TP: True positive.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-021-04041-7.

Additional file 1. Supplementary materials providing technical details about quality control of genetic data, theory, implementation as well as figures.

Acknowledgements

We thank the Gemini Center for Sepsis Research for enabling contact with colleagues at Yale School of Public Health. We thank the Yale Center for Research Computing for guidance and use of the research computing infrastructure.

Authors' contributions

PVJ designed, with guidance from SRS, ATD, ML and MEC, developed and implemented the software. The results were analyzed by PVJ with guidance from SRS, ATD, ML and MEC. MEC implemented and provided the results from the BOLT-LMM software. PVJ wrote the manuscript and the Additional File 1. SRS, ATD, ML and MEC edited the manuscript and the Additional File 1. All authors read and approved the final manuscript.

Funding

This research was funded by the The Research Council of Norway, Grant 272402, PhD Scholarship at SINTEF, including funding for research stay abroad at Yale School of Public Health.

Availability of data and materials

The research has been conducted using the UK Biobank Resource under Application Number 32285. The application for access to the UK Biobank Resource was approved on October 10, 2018. Source code supporting this paper can be found online at https://github.com/palVJ/GWASwithTreeSHAP/.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

SINTEF DIGITAL is an institute at SINTEF. SINTEF is a non-profit research organization that provided the PhD Scholarship for first author via The Research Council of Norway. Neither The Research Council of Norway nor SINTEF DIGITAL had any role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. The authors declare that they have no competing interests.

Author details

[1]SINTEF DIGITAL, Forskningsveien 1, 0373 Oslo, Norway. [2]Department of Mathematical Sciences, Norwegian University of Science and Technology, A. Getz vei 1, 7491 Trondheim, Norway. [3]Department of Chronic Disease Epidemiology and Center for Perinatal, Pediatric and Environmental Epidemiology, Yale School of Public Health, 1 Church Street, New Haven, CT 06510, USA. [4]Gemini Center for Sepsis Research, Department of Circulation and Medical Imaging, NTNU, Norwegian University of Science and Technology, Prinsesse Kristinas gate 3, 7030 Trondheim, Norway.

Received: 3 June 2020 Accepted: 22 February 2021
Published online: 04 May 2021

References

1. Loh PR, Tucker G, Bulik-Sullivan BK, Vilhjálmsson BJ, Finucane HK, Salem RM, et al. Efficient Bayesian mixed model analysis increases association power in large cohorts. Nat Genet. 2015;47(3):284–90.
2. Visscher PM, Wray NR, Zhang Q, Sklar P, McCarthy MI, Brown MA, et al. 10 years of GWAS discovery: biology, function, and translation. Am J Human Genet. 2017;101(1):5–22.
3. Yu J, Pressoir G, Briggs WH, Vroh Bi I, Yamasaki M, Doebley JF, et al. A unified mixed-model method for association mapping that accounts for multiple levels of relatedness. Nat Genet. 2006;38(2):203–8.
4. Maher B, Maher B, editor. Personal genomes: the case of the missing heritability [News]. Nature. 2008.
5. Burton PR, Hansell AL, Fortier I, Manolio TA, Khoury MJ, Little J, et al. Size matters: just how big is BIG? Quantifying realistic sample size requirements for human genome epidemiology. Int J Epidemiol. 2009.
6. Langaas M, Bakke Ø. Robust methods to detect disease-genotype association in genetic association studies: calculate p values using exact conditional enumeration instead of simulated permutations or asymptotic approximations. Stat Appl Genet Mol Biol. 2014;13(6):675–92.
7. Cordell HJ. Epistasis: what it means, what it doesn't mean, and statistical methods to detect it in humans. Hum Mol Genet. 2002;11:2463–8.
8. Phillips PC. Epistasis-the essential role of gene interactions in the structure and evolution of genetic systems. Nat Rev Genet. 2008;9(11):855–67.
9. Ritchie MD, Steen KV. The search for gene-gene interactions in genome-wide association studies: challenges in abundance of methods, practical considerations, and biological interpretation. Ann Transl Med. 2018;6(8):21–21.
10. Yung LS, Yang C, Wan X, Yu W. GBOOST: a GPU-based tool for detecting gene-gene interactions in genome-wide case control studies. Bioinformatics. 2011 May;27.
11. Wan X, Yang C, Yang Q, Xue H, Fan X, Tang NLS, et al. BOOST: a fast approach to detecting gene-gene interactions in genome-wide case-control studies. Am J Hum Genet. 2010;87:325–40.
12. Hu X, Liu Q, Zhang Z, Li Z, Wang S, He L, et al. SHEsisEpi, a GPU-enhanced genome-wide SNP-SNP interaction scanning algorithm, efficiently reveals the risk genetic epistasis in bipolar disorder. Cell Research. 2010;20:854–7.
13. Goudey B, Rawlinson D, Wang Q, Shi F, Ferra H, Campbell RM, et al. GWIS - model-free, fast and exhaustive search for epistatic interactions in case-control GWAS. BMC Genom. 2013;14:S10.
14. Chatelain C, Durand G, Thuillier V, Augé F. Performance of epistasis detection methods in semi-simulated GWAS. BMC Bioinform. 2018;19:231.
15. Li D, Won S. Efficient strategy to identify gene–gene interactions and its application to Type 2 diabetes. Genom Inform. 2016;14:160–5.
16. Chen T, Guestrin C. XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining - KDD '16. 2016;p. 785–794.
17. Lundberg SM, Erion G, Chen H, DeGrave A, Prutkin JM, Nair B, et al. From local explanations to global understanding with explainable AI for trees. Nat Mach Intell. 2020;2(1):56–67.
18. Locke AE, Kahali B, Berndt SI, Justice AE, Pers TH, Day FR, et al. Genetic studies of body mass index yield new insights for obesity biology. Nature. 2015;518(7538):197–206 (Number: 7538 Publisher: Nature Publishing Group.).
19. Speliotes EK, Willer CJ, Berndt SI, Monda KL, Thorleifsson G, Jackson AU, et al. Association analyses of 249,796 individuals reveal 18 new loci associated with body mass index. Nat Genet. 2010;42(11):937–48.
20. Thorleifsson G, Walters GB, Gudbjartsson DF, Steinthorsdottir V, Sulem P, Helgadottir A, et al. Genome-wide association yields new sequence variants at seven loci that associate with measures of obesity. Nat Genet. 2009;41(1):18–24 (Number: 1 Publisher: Nature Publishing Group.).
21. Schwarz DF, König IR, Ziegler A. On safari to Random Jungle: a fast implementation of Random Forests for high-dimensional data. Bioinformatics. 2010;26:1752–8.

22. Lubke G, Laurin C, Walters R, Eriksson N, Hysi P, Spector T, et al. Gradient boosting as a SNP filter: an evaluation using simulated and hair morphology data. J Data Min Genom Proteom. 2013;4.
23. Yin B, Balvert M, van der Spek RAA, Dutilh BE, Bohté S, Veldink J, et al. Using the structure of genome data in the design of deep neural networks for predicting amyotrophic lateral sclerosis from genotype. Bioinformatics. 2019;35:i538-47.
24. Romagnoni A, Jégou S, Van Steen K, Wainrib G, Hugot JP. Comparative performances of machine learning methods for classifying Crohn Disease patients using genome-wide genotyping data. Sci Rep. 2019;9:1–18.
25. Friedman JH. Greedy function approximation: a gradient boosting machine. Ann Stat. 2001;29:1189–232.
26. Hastie T, Tibshirani R, Friedman J. The elements of statistical learning. 2nd ed. Springer; 2009.
27. Nielsen D. Tree boosting With XGBoost. Norwegian University of Science and Technology; 2016.
28. Alsabti K, Ranka S, Singh V. CLOUDS: A decision tree classifier for large datasets. In: Agrawal R, Stolorz P, editors. Proceedings of the 4th knowledge discovery and data mining conference; 1998. p. 2–8.
29. Jin R, Agrawal G. Communication and memory efficient parallel decision tree construction. In: Barbara D, Kamath C, editors. Proceedings of the 2003 SIAM international conference on data mining; 2003. p. 119–129.
30. Li P, Wu Q, Burges CJ. McRank: learning to rank using multiple classification and gradient boosting. In: Platt JC, Koller D, Singer Y, Roweis ST, editors. Advances in neural information processing systems 20. Curran Associates: Inc; 2008. p. 897–904.
31. Blagus R, Lusa L. Boosting for high-dimensional two-class prediction. BMC Bioinform. 2015;16.
32. Grau J, Grosse I, Keilwagen J. PRROC: computing and visualizing precision-recall and receiver operating characteristic curves in R. Bioinformatics. 2015;31(15):2595–7.
33. Breiman L, Friedman J, Stone CJ, Olshen RA. Classification and regression trees. Taylor & Francis; 1984.
34. Lundberg SM, Erion GG, Lee S. Consistent Individualized feature attribution for tree ensembles. CoRR. 2018;Available from: arxiv:1802.03888.
35. Lundberg SM, Lee SI. A unified approach to interpreting model predictions. In: Advances in neural information processing systems; 2017. p. 4765–4774.
36. Roth AE. The Shapley value: Essays in honor of Lloyd S.Shapley. Cambridge University Press. 1998;p. 10.
37. Janzing D, Minorics L, Blöbaum P. Feature relevance quantification in explainable AI: A causal problem. arXiv:191013413 [cs, stat]. 2019;.
38. Aas K, Jullum M, Løland A. Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. arXiv:190310464 [cs, stat]. 2019 Jun;.
39. Fujimoto K, Kojadinovic I, Marichal JL. Axiomatic characterizations of probabilistic and cardinal-probabilistic interaction indices. Games Econ Behav. 2006;55(1):72–99.
40. Ambroise C, McLachlan GJ. Selection bias in gene extraction on the basis of microarray gene-expression data. Proc Natl Acad Sci. 2002;99(10):6562–6.
41. Reich DE, Cargill M, Bolk S, Ireland J, Sabeti PC, Richter DJ, et al. Linkage disequilibrium in the human genome. Nature. 2001;411:199–204.
42. Sillanpää MJ. Overview of techniques to account for confounding due to population stratification and cryptic relatedness in genomic data association analyses. Heredity. 2011;106(4):511–9.
43. McLachlan GJ, Chevelu J, Zhu J. Correcting for selection bias via cross-validation in the classification of microarray data. Institute of Mathematical Statistics. 2008;.
44. Bycroft C, Freeman C, Petkova D, Band G, Elliott LT, Sharp K, et al. Genome-wide genetic data on 500,000 UK Biobank participants. bioRxiv. 2017;p. 166298.
45. Bjørnland T, Bye A, Ryeng E, Wisløff U, Langaas M. Powerful extreme phenotype sampling designs and score tests for genetic association studies. Stat Med. 2018;37(28):4234–51.
46. Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira MAR, Bender D, et al. PLINK: a tool set for whole-genome association and population-based linkage analyses. Am J Hum Genet. 2007;81(3):559–75.
47. Chen T, He T, Benesty M, Khotilovich V, Tang Y, Cho H, et al.. xgboost: Extreme Gradient Boosting; 2019. R package version 1.0.0.2. Available from: https://CRAN.R-project.org/package=xgboost.
48. Phillips CM, Kesse-Guyot E, McManus R, Hercberg S, Lairon D, Planells R, et al. High dietary saturated fat intake accentuates obesity risk associated with the fat mass and obesity-associated gene in adults. J Nutr. 2012;142(5).
49. Pietiläinen KH, Kaprio J, Borg P, Plasqui G, Yki-Järvinen H, Kujala UM, et al. Physical inactivity and obesity: a vicious circle. Obesity (Silver Spring, Md). 2008;16(2):409–14.
50. Lourenço S, Oliveira A, Lopes C. The effect of current and lifetime alcohol consumption on overall and central obesity. Eur J Clin Nutr. 2012;66(7):813–8.
51. Scott KA, Melhorn SJ, Sakai RR. Effects of chronic social stress on obesity. Curr Obes Rep. 2012;1(1):16–25.
52. Cappuccio FP, Taggart FM, Kandala NB, Currie A, Peile E, Stranges S, et al. Meta-analysis of short sleep duration and obesity in children and adults. Sleep. 2008;31(5):619–26.
53. Loh PR, Kichaev G, Gazal S, Schoech AP, Price AL. Mixed-model association for biobank-scale datasets. Nat Genet. 2018;50:906–8.
54. Yu Z, Demetriou M, Gillen DL. Genome-wide analysis of gene-gene and gene-environment interactions using closed-form wald tests. Genet Epidemiol. 2015;39:446–55.
55. Galinsky K, Bhatia G, Loh PR, Georgiev S, Mukherjee S, Patterson N, et al. Fast principal-component analysis reveals convergent evolution of ADH1B in Europe and East Asia. Am J Hum Genet. 2016;98:456–72.
56. Price AL, Patterson NJ, Plenge RM, Weinblatt ME, Shadick NA, Reich D. Principal components analysis corrects for stratification in genome-wide association studies. Nat Genet. 2006;38(8):904–9.
57. Patterson N, Price AL, Reich D. Population structure and eigenanalysis. PLoS Genet. 2006;2(12):e190.
58. Goeman JJ, Solari A. Multiple hypothesis testing in genomics. Stat Med. 2014;33(11):1946–78.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

A new method for exploring gene-gene and gene-environment interactions in GWAS with tree ensemble methods and SHAP values
Supplementary File

Pål Vegard Johnsen[1,2], Signe Riemer-Sørensen[1], Andrew Thomas DeWan[3], Megan E. Cahill[3], and Mette Langaas[2]

[1]SINTEF Digital, Oslo, Norway
[2]Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway
[3]Department of Chronic Disease Epidemiology and Center for Perinatal, Pediatric and Environmental Epidemiology, Yale School of Public Health

1 Quality assessment of UK Biobank Genetic Data

Analyses were limited to autosomal variants covered by both genotype arrays used over the course of the study and that passed the batch-level quality control. SNPs were included if the call rate was above 99%, the Hardy-Weinberg equilibrium p-value was less than $5 \cdot 10^{-8}$, and the minor allele frequency was larger than 1%. 529,024 SNPs passed these filters.

Individuals were removed if the genetic and reported sex did not match and if the sex chromosomes were not XX or XY. Outliers in heterozygosity and missing rates were removed. The analyses were limited to those identified as Caucasian through the UK Biobank's PCA analysis (field 22006). All individuals had an individual call rate larger than 99%. 366,752 individuals passed these filters.

2 Details of environmental features from UK Biobank

A sample set of personal and environmental characteristics were included in the model as features to demonstrate sample use of the method. All descriptions are from the UK Biobank Showcase, and no outliers were removed. Individuals that answered "prefer not to answer" or "do now know" to any given question were treated as missing values. All features are taken from the baseline assessment, the same point in time when the BMI phenotype was measured. The following environmental and personal features collected at baseline were evaluated:

Description	**Data field**
Age when attended assessment centre	21003
Genetic sex	22001
Number of days/week walked 10+ minutes	864
Minutes spent walking per day	874
Number of days/week of moderate physical activity 10+ minutes	884
Duration of moderate activity per day	894
Number of days/week of vigorous physical activity 10+ minutes	904
Duration of vigorous activity per day	914
Alcohol intake frequency	1558
Sleep duration	1160
Processed meat intake	1349
Beef intake	1369
Lamb/mutton intake	1379
Pork intake	1389
Cheese intake	1408
Milk type used	1418
Illness, injury, bereavement, stress in last 2 years	6145

2.1 Age when attended assessment centre

Age at the initial assessment visit (2006-2010) during which participants were recruited and provided consent.

2.2 Genetic sex

Sex as determined from genotyping analysis.

2.3 Physical activity

To measure the degree of physical activity, the duration of walking, moderate activity and vigorous activity per day were added with equal weight. The duration of any given activity per day is set to zero if an individual spent no days during the week with more than 10 minutes of that activity.

2.4 Alcohol intake

Participants were asked how frequently they consumed alcohol, with potential responses never, only on special occasions, one to three times a month, one to three times a week, three or four times a week, or daily or nearly daily.

2.5 Sleep duration

Participants were asked to report how many hours of sleep they got in a 24 hour period.

2.6 Saturated fat intake

Participants were asked how frequently they consumed each food item, from never to daily. Frequency of beef, lamb, mutton, pork, cheese and milk intake per week was added with equal weight.

2.7 Stressful events

We treated this as a binary variable, such that those that have not experienced any of the categories listed in the ”Illness, injury, bereavement, stress in last 2 years” variable during the past two years are represented by the value zero, and the rest were set to one.

2.8 Treatment of categorical features and correlation plot

XGBoost does not automatically take into account categorical features. Sex, alcohol consumption and sleep duration can be considered categorical features, but as sex is a binary feature, while alcohol consumption and sleep duration are ordinal features, a split between two categories for these features in a regression tree is meaningful, and therefore the features can be treated as they are. The correlation of the final seven environmental features were investigated further by computing the Pearson's correlation between all pairs of features by excluding missing values. No pair of features showed Pearson's correlation r larger than 0.2, and we therefore treat these features as if they were independent of each other when computing the SHAP values. Correlations between environmental features and SNPs are also surprisingly not very small. Even though there exist dependence between SNPs and environmental features, the effects are so small that we also in this case regard them to be independent to each other when computing the SHAP values.

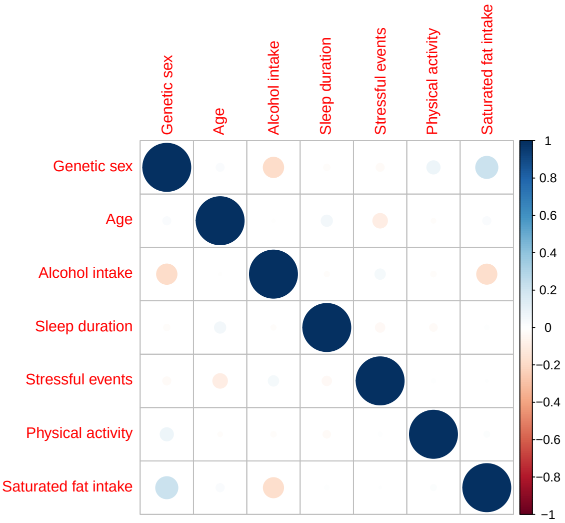


Figure 1: Pearson's correlation, r, between environmental features.

3 The minimum number of random subsets to choose in the ranking process

In Phase 1 of the method described in the main article, we perform a ranking process for the SNPs using a combination of random subsets of SNPs with cross-validation. Here we show the probability calculations guiding the choice of the number of random subsets of SNPs that we use, first for one SNP and then for a SNP pair.

3.1 Number of subsets for single SNP sampling

We have a total of R SNPs, and draw $S < R$ SNPs without replacement. Let $A = 1$ denote the case where we study one randomly sampled subset of S SNPs, and $A = a$ the case where we study a different samples. The question is how large a at least should be in order to investigate the whole genome to a sufficient extent.

Let C_j be the number of times a particular SNP j is chosen among all $A = a$ subsets. Since the SNPs are randomly sampled without replacement, the probability that SNP j is contained in at least one of the a subsets, $P(C_j \geq 1|A = a)$, is given by:

$$P(C_j \geq 1|A = a) = 1 - P(C_j = 0|A = a) = 1 - P(C_j = 0|A = 1)^a = 1 - \big(1 - \frac{S}{R}\big)^a,$$

since $P(C_j = 0|A = 1)$ is given from the corresponding hypergeometric distribution:

$$P(C_j = 0|A = 1) = \frac{\binom{1}{0}\binom{R-1}{S}}{\binom{R}{S}} = 1 - \frac{S}{R}.$$

If we want the probability to be larger than some preferred value p, we get the inequality referred to in the main article:

$$a \geq \frac{\log(1-p)}{\log(1-\frac{S}{R})}. \tag{1}$$

However, after the SNPs are randomly sampled, we also perform a pruning to minimize the correlation in the sample as explained in Section 4, so the number of subsets to create should be even larger than this.

3.2 Number of subsets for pair SNP sampling

Similarly, assume the SNPs to be randomly sampled, and let $C_{j,k}$ be the number of times SNP j and SNP k are present simultaneously in a total of a subsets. We then have:

$$\begin{aligned} P(C_{j,k} \geq 1|A = a) &= 1 - P(C_{j,k} = 0|A = a) = 1 - P(C_{j,k} = 0|A = 1)^a \\ &= 1 - \big(1 - P(C_{j,k} = 1|A = 1)\big)^a \\ &= 1 - \left(1 - \frac{S(S-1)}{R(R-1)}\right)^a, \end{aligned}$$

since $P(C_{j,k} = 1|A = 1)$ is given from a corresponding hypergeometric distribution:

$$P(C_{jk} = 1 | A = 1) = \frac{\binom{2}{2}\binom{R-2}{S-2}}{\binom{R}{S}} = \frac{S(S-1)}{R(R-1)}.$$

For this probability to be larger than a preferred value p, we get the inequality referred to in the main article:

$$a \geq \frac{\log(1-p)}{\log\left(\frac{S(S-1)}{R(R-1}\right)}. \tag{2}$$

Again, the total number of subsets should be larger due to the need for SNP pruning to ensure low correlation among the SNPs. Anyhow, inequalities (1) and (2) can be used as guidance as to how many subsets should at least be created.

4 SNP pruning with PLINK1.9

When creating the subsets explained in Section 3.1 (the ranking process) of the main article, we create a subset of S SNPs with mutually low correlation together with G randomly sampled individuals. This is implemented by using both R and PLINK1.9 [4].

First, S^* SNPs and G individuals are sampled with equal probability and without replacement. Next we apply the PLINK1.9 function −−`indep-pairwise` with the following parameter values window size = 50 kb, step size = 5kb and $r^2 = 0.2$ in order to get a subset of S SNPs were all pairs of SNPs within a region of 50 kilobases have squared Pearson's correlation less than 0.2. SNPs that are more than 50 kilobases from each other are not expected to correlate to any significant extent. Pearson correlation measures linear dependency, and therefore zero correlation does not imply independence in general. We will anyhow rely on r^2 as a measure of independence due to its fast computation on large amounts of data. In the example analysis we manually find, by trial and error, the appropriate size of S^* corresponding to the chosen value for S.

In a similar manner, the PLINK1.9 function −−`indep-pairwise` can be used to obtain a subset of SNPs with mutually low correlation based on some ranked set of SNPs, as in Section 3.2 (model fitting process) in the main article. However, the ranked list of SNPs should be added as a .frq-datafile via −−read-freq, where the column variable MAF is edited such that it does not denote the minor allele frequencies, but some feature importance score of each SNP. The larger the score is, the higher priority the SNP will have to be kept among the subset.

5 Running BOLT-LMM on the ranking data

In the obesity example, we run BOLT-LMM on the ranking data (from Phase 1) with obesity as trait in order to rank the importance of each SNP based on the their computed p-values by using the BOLT-LMM-infinitesimal mixed-model statistic [2]. BOLT-LMM is intentionally constructed for quantitative traits and not for case-control traits such as obesity, but it can be applied by treating the binary trait as a quantitative trait. The caveat is however that the p-values may be invalid. However, the p-values computed have been shown to be valid as long as the MAFs of each SNP are larger than 1%, and that the case fraction is larger than 30% for a sample of 50 000 individuals [2]. The ranking data has a case fraction of 43 %, MAF greater than 1 % and 80 000 individuals, and so we regard the p-values computed as valid. Obesity and features were defined as described in Appendix B in the main article. Categorical features in the model were genetic sex, alcohol intake frequency, sleep duration (in hours), and any events of illness, injury, bereavement, or stress in the previous two years. Quantitative features were physical activity, saturated fat intake, and age at initial assessment. All features excluding genetic sex were self-reported during the initial assessment.

6 Computations of SHAP values

The SHAP value, $\phi_{i,j}(\mathbf{x}_i)$, for a model $f(\mathbf{x}_i)$, individual i and feature j given all features $\mathbf{x}_i$ is defined in Lundberg et al. [3] and Janzing, Minorics, and Blöbaum [1] as:

$$\phi_{i,j}(\mathbf{x}_i) = \textstyle\sum_{\mathcal{S}\subseteq\mathcal{M}\setminus\{j\}} \frac{|\mathcal{S}|!(M-|\mathcal{S}|-1)!}{M!} \left[E[f(\mathbf{X}_{i,\mathcal{S}\cup\{j\}} = \mathbf{x}^*_{i,\mathcal{S}\cup\{j\}}, \mathbf{X}_{i,\overline{\mathcal{S}\cup\{j\}}})] - E[f(\mathbf{X}_{i,\mathcal{S}} = \mathbf{x}^*_{i,\mathcal{S}}, \mathbf{X}_{i,\overline{\mathcal{S}}})]\right] \tag{3}$$

where $E[f(\mathbf{X}_{i,\mathcal{S}\cup\{j\}} = \mathbf{x}^*_{i,\mathcal{S}\cup\{j\}}, \mathbf{X}_{i,\overline{\mathcal{S}\cup\{j\}}})]$ is the expected prediction when only the values of the feature subset $\mathcal{S}$ as well as feature j, denoted $\mathbf{x}^*_{i,\mathcal{S}\cup\{j\}}$, are known, while the vector of unknown values from the complement set, $\mathbf{X}_{i,\overline{\mathcal{S}\cup\{j\}}}$ are regarded as a random vector. Notice that $S\cup\overline{S} = \mathcal{M}$.

6.1 SHAP values for tree ensemble models

We consider a tree ensemble model where the prediction, $f(\mathbf{x}_i)$, is a linear sum of outputs from all regression trees given features $\mathbf{x}_i$. By the linearity property of expectation, the marginal expectation, $E[f(\mathbf{X}_{i,\mathcal{S}} = \mathbf{x}^*_{i,\mathcal{S}}, \mathbf{X}_{i,\overline{\mathcal{S}}})]$, given in Equation (3) is equal to the sum of the marginal expectation of the output from each regression tree, denoted $E[f_\tau(\mathbf{X}_{i,\mathcal{S}} = \mathbf{x}^*_{i,\mathcal{S}}, \mathbf{X}_{i,\overline{\mathcal{S}}})]$:

$$E[f(\mathbf{X}_{i,\mathcal{S}} = \mathbf{x}^*_{i,\mathcal{S}}, \mathbf{X}_{i,\overline{\mathcal{S}}})] = \sum_{\tau=1}^{T} E[f_\tau(\mathbf{X}_{i,\mathcal{S}} = \mathbf{x}^*_{i,\mathcal{S}}, \mathbf{X}_{i,\overline{\mathcal{S}}})].$$

The marginal expectation for each regression tree, assuming only continuous features, is mathematically expressed as:

$$E[f_\tau(\mathbf{X}_{i,\mathcal{S}} = \mathbf{x}^*_{i,\mathcal{S}}, \mathbf{X}_{i,\overline{\mathcal{S}}})] = \int_{\mathbf{x}_{i,\overline{\mathcal{S}}}} f_\tau(\mathbf{X}_{i,\mathcal{S}} = \mathbf{x}^*_{i,\mathcal{S}}, \mathbf{X}_{i,\overline{\mathcal{S}}} = \mathbf{x}^*_{i,\overline{\mathcal{S}}}) p(\mathbf{X}_{i,\bar{\mathcal{S}}} = \mathbf{x}^*_{i,\overline{\mathcal{S}}}) d\mathbf{x}_{i,\overline{\mathcal{S}}}, \tag{4}$$

where we denote $\mathbf{x}^*_i = (\mathbf{x}^*_{i,\mathcal{S}}, \mathbf{x}^*_{i,\bar{\mathcal{S}}})$ as the constant vector where all feature values are known. As each regression tree f_τ only takes a distinct number of values equal to the number of leaves B_τ in the regression tree, the integral in (4) can be expressed as a sum of integrals:

$$E[f_\tau(\mathbf{X}_{i,\mathcal{S}} = \mathbf{x}^*_{i,\mathcal{S}}, \mathbf{X}_{i,\overline{\mathcal{S}}})] = \sum_{k=1}^{B_\tau} c_{\tau,k} \int_{\mathbf{x}_{i,\overline{\mathcal{S}_{\tau,k}}}} p(\mathbf{X}_{i,\overline{\mathcal{S}}} = \mathbf{x}^*_{i,\overline{\mathcal{S}_{\tau,k}}}) d\mathbf{x}_{i,\overline{\mathcal{S}_{\tau,k}}},$$

where each $\mathbf{x}^*_{i,\overline{\mathcal{S}_{\tau,k}}}$ is such that $f_\tau(\mathbf{x}^*_i = (\mathbf{x}^*_{i,\mathcal{S}}, \mathbf{x}^*_{i,\overline{\mathcal{S}_{\tau,k}}})) = c_{\tau,k}$ where $c_{\tau,k}$ is leaf value number k for tree τ.

If we assume the complement subset $\bar{\mathcal{S}}$ of features are mutually independent, the integral can be further partitioned into a product of integrals , where each integral will be integrated over the range of the corresponding feature in $\bar{\mathcal{S}}$ that leads to the path from root to leaf node with leaf node value $c_{\tau k}$:

$$E[f_\tau(\mathbf{X}_{i,\mathcal{S}} = \mathbf{x}^*_{i,\mathcal{S}}, \mathbf{X}_{i,\overline{\mathcal{S}}})] = \sum_{k=1}^{B_\tau} c_{\tau,k} \prod_{\ell=1}^{l} \int_{x_{i,\ell}=a_{\ell,\tau,k}}^{b_{\ell,\tau,k}} p(X_{i,\ell} = x^*_{i,\ell}) dx_{i,\ell},$$

where $x_{i,\ell}$ denotes the feature value of feature number ℓ among a total of l unknown features in the subset $\bar{\mathcal{S}}$, while $(a_{\ell,\tau,k}, b_{\ell,\tau,k})$ is the range in which feature number ℓ must be integrated over in order to get the output value $c_{\tau,k}$ for regression tree τ. For features in $\bar{\mathcal{S}}$ that are not present in the regression tree τ, these features can take any value. We define the value of the corresponding integrals in the product operator to be one.

What remains in order to compute the marginal expectation given in Equation (3) is to estimate each of the integrals given above. In Lundberg et al. [3] these are estimated by using the proportion of samples in each node in each tree in the training phase of the tree ensemble model that goes in the same direction from a particular node to another. Under the assumption of mutual independence this is a reasonable estimate, but the estimate naturally relies on the total number of individuals that are used for estimation, and so these estimations will be poorer the deeper the trees are. Finally, and most importantly, in order to compute the SHAP values for a tree ensemble model, Lundberg et al. [3] have constructed an algorithm with polynomial running time, $O(TLD^2)$, for maximum depth D and leaves L.

7 Logistic regression with different additivity assumptions

In the main article, all likelihood ratio tests are based on the assumption of both additive marginal effects and additive interaction effects. Here we provide two additional tests with less stricter additive assumptions.

For the case of SNP-SNP interactions, the first model is unconstrained in both main effects and interactions [5]:

$$\begin{aligned}
&\mathrm{logit}(P(Y_i = 1|g_{i,a}, g_{i,b}, \mathbf{x}_{i,c})) = \\
&\mathbf{x}_{i,c}^T\gamma + \alpha_1 I(g_{i,a} = 1) + \alpha_2 I(g_{i,a} = 2) + \beta_1 I(g_{i,b} = 1) + \beta_2 I(g_{i,b} = 2) \\
&+ \nu_{11} I(g_{i,a} = 1) I(g_{i,b} = 1) + \nu_{12} I(g_{i,a} = 1) I(g_{i,b} = 2) \\
&+ \nu_{21} I(g_{i,a} = 2) I(g_{i,b} = 1) + \nu_{22} I(g_{i,a} = 2) I(g_{i,b} = 2),
\end{aligned} \tag{5}$$

where $\mathbf{x}_{i,c}^T$ is a vector of features such as intercept, age, environmental features and principal components, γ is the vector of corresponding parameters for each feature, $I()$ is the indicator function, α_1, α_2, β_1 and β_2 are marginal effects of the SNPs $g_{i,a}$ and $g_{i,b}$ when the genotype value is one or two respectively, while ν_{11}, ν_{12}, ν_{21} and ν_{22} are unconstrained interaction parameters for $g_{i,a}$ and $g_{i,b}$.

When testing the presence of interaction effects, the null hypothesis is $\nu_{11} = \nu_{12} = \nu_{21} = \nu_{22} = 0$, with null model:

$$\begin{aligned}
&\mathrm{logit}_{H_0}(P(Y_i = 1|g_{i,a}, g_{i,b}, \mathbf{x}_{i,c})) = \\
&\mathbf{x}_{i,c}^T\gamma + \alpha_1 I(g_{i,a} = 1) + \alpha_2 I(g_{i,a} = 2) + \beta_1 I(g_{i,b} = 1) + \beta_2 I(g_{i,b} = 2).
\end{aligned} \tag{6}$$

If we assume additive interaction effects, corresponding to $\nu_{11} = \nu$, $\nu_{12} = \nu_{21} = 2\nu$ and $\nu_{22} = 4\nu$, we get the alternative model:

$$\begin{aligned}
&\mathrm{logit}(P(Y_i = 1|g_{i,a}, g_{i,b}, \mathbf{x}_{i,c})) = \mathbf{x}_{i,c}^T\gamma + \alpha_1 I(g_{i,a} = 1) + \alpha_2 I(g_{i,a} = 2) \\
&+ \beta_1 I(g_{i,b} = 1) + \beta_2 I(g_{i,b} = 2) + \nu g_{i,a} g_{i,b}.
\end{aligned} \tag{7}$$

We will then have two new tests based on the following null and alternative models: Models (6) and (5) in the case of no assumptions and models (6) and (7) in the case of additive interactions. We denote these tests as Test 1 and Test 2 respectively. The test applied in the main article is denoted as Test 3 with null and alternative models:

$$\mathrm{logit}_{H_0,add}(P(Y_i = 1|g_{i,a}, g_{i,b}, \mathbf{x}_{i,c})) = \mathbf{x}_{i,c}^T\gamma + \alpha g_{i,a} + \beta g_{i,b}. \tag{8}$$

$$\mathrm{logit}_{H_1,add}(P(Y_i = 1|g_{i,a}, g_{i,b}, \mathbf{x}_{i,c})) = \mathbf{x}_{i,c}^T\gamma + \alpha g_{i,a} + \beta g_{i,b} + \nu g_{i,a} g_{i,b}. \tag{9}$$

For the case of SNP-environment interactions, the logistic models will look similar in the case where the environmental feature is discrete. For the case where the environmental feature, $x_{i,e}$, is continuous, the unconstrained Test 1 will for instance have the following alternative model:

$$\begin{aligned}
\mathrm{logit}(P(Y_i = 1|g_{i,a}, x_{i,e}, \mathbf{x}_{i,c})) = \mathbf{x}_{i,c}^T\gamma &+ \alpha_1 I(g_{i,a} = 1) + \alpha_2 I(g_{i,a} = 2) + \beta_e x_{i,e} \\
&+ \phi_1 I(g_{i,a} = 1) x_{i,e} + \phi_2 I(g_{i,a} = 2) x_{i,e},
\end{aligned} \tag{10}$$

where β_e, ϕ_1 and ϕ_2 are the marginal effect of the environmental feature, and interaction effects respectively.

The results when applying all three tests for each of the interactions based on both the evaluation data and all individuals is given in Table 1.

Table 1: Results from all likelihood ratio tests with different assumptions of additivity. The tests are applied on the top four ranked interactions found from the model explainability process based on the evaluation data.

Test	**Interaction**	***p*-value LRT**
Test 1 evaluation data	rs171329 and rs180743	0.49
Test 1 all individuals	rs171329 and rs180743	0.0063
Test 2 evaluation data	rs171329 and rs180743	0.85
Test 2 all individuals	rs171329 and rs180743	0.024
Test 3 evaluation data	rs171329 and rs180743	0.85
Test 3 all individuals	rs171329 and rs180743	0.024
Test 1 evaluation data	rs17817449 and genetic sex	0.96
Test 1 all individuals	rs17817449 and genetic sex	0.00022
Test 2 evaluation data	rs17817449 and genetic sex	0.79
Test 2 all individuals	rs17817449 and genetic sex	4.78e-05
Test 3 evaluation data	rs17817449 and genetic sex	0.77
Test 3 all individuals	rs17817449 and genetic sex	4.09e-05
Test 1 evaluation data	rs17817449 and saturated fat intake	0.59
Test 1 all individuals	rs17817449 and saturated fat intake	0.0019
Test 2 evaluation data	rs17817449 and saturated fat intake	0.45
Test 2 all individuals	rs17817449 and saturated fat intake	0.0017
Test 3 evaluation data	rs17817449 and saturated fat intake	0.44
Test 3 all individuals	rs17817449 and saturated fat intake	0.0017
Test 1 evaluation data	rs757318 and rs12123815	0.48
Test 1 all individuals	rs757318 and rs12123815	0.49
Test 2 evaluation data	rs757318 and rs12123815	0.25
Test 2 all individuals	rs757318 and rs12123815	0.71
Test 3 evaluation data	rs757318 and rs12123815	0.25
Test 3 all individuals	rs757318 and rs12123815	0.71

Even though the three statistical tests have different assumptions, the *p*-values for the three tests for each interaction do not vary greatly. Therefore, in this case, the assumptions of additivity do not have any significant impact of the computed p-values.

8 PCA plots - Evaluation data and full dataset

Figure 2: PCA plot for the first and second principal components for unrelated individuals in the full dataset.

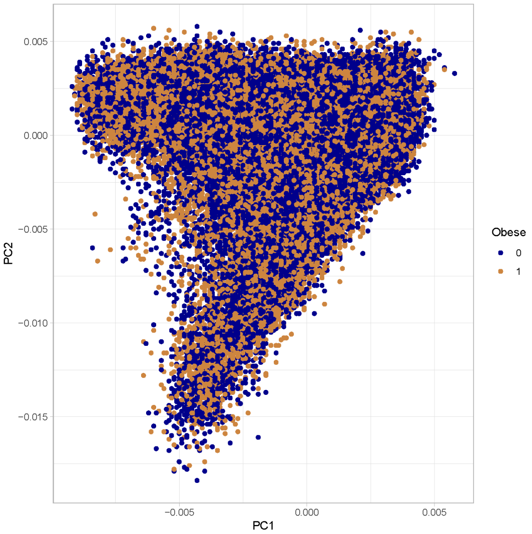


Figure 3: PCA plot for third and fourth principal components for unrelated individuals in the full dataset.

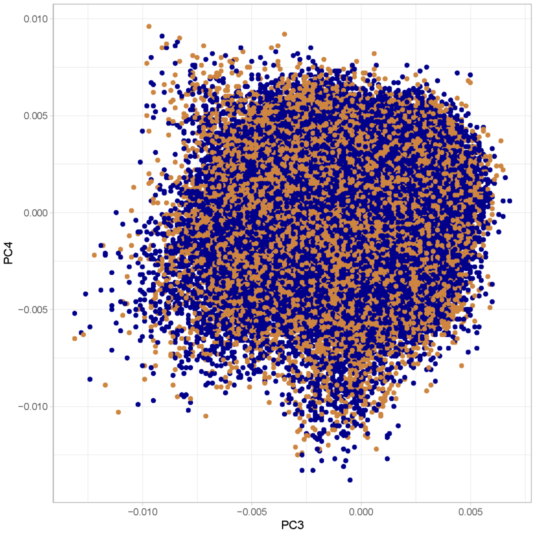


Figure 4: PCA plot for first and second principal components for unrelated individuals in the evaluation dataset.

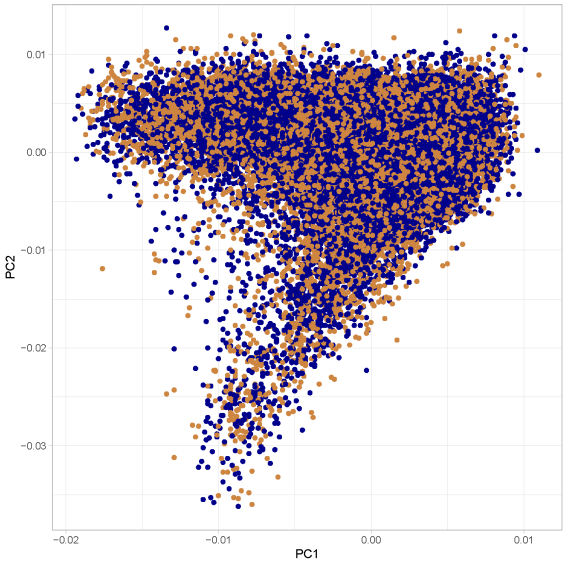
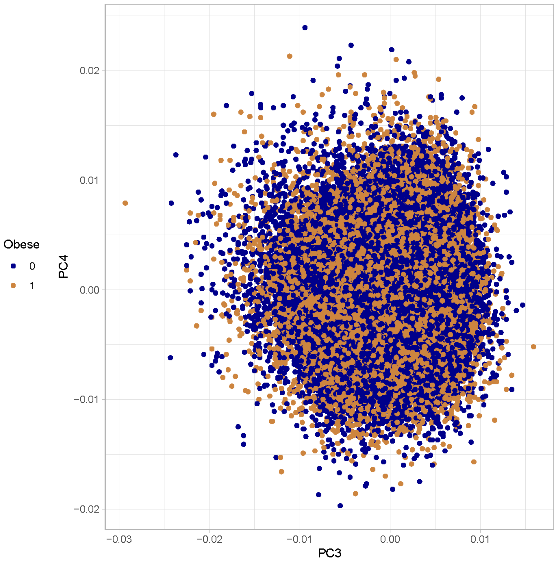


Figure 5: PCA plot for third and fourth principal components for unrelated individuals in the evaluation dataset.



References

[1] Dominik Janzing, Lenon Minorics, and Patrick Blöbaum. “Feature relevance quantification in explainable AI: A causal problem”. In: *arXiv:1910.13413 [cs, stat]* (2019).

[2] Po-Ru Loh et al. “Mixed-model association for biobank-scale datasets”. In: *Nature Genetics* 50 (July 2018), pp. 906–908.

[3] Scott M. Lundberg et al. “From local explanations to global understanding with explainable AI for trees”. In: *Nature Machine Intelligence* 2.1 (Jan. 2020), pp. 56–67.

[4] Shaun Purcell et al. “PLINK: A Tool Set for Whole-Genome Association and Population-Based Linkage Analyses”. In: *American Journal of Human Genetics* 81.3 (2007), pp. 559–575.

[5] Zhaoxia Yu, Michael Demetriou, and Daniel L. Gillen. “Genome-Wide Analysis of Gene-Gene and Gene-Environment Interactions Using Closed-Form Wald Tests”. In: *Genetic Epidemiology* 39.6 (2015).

Paper 3

Inferring feature importance with uncertainties in high-dimensional data

Inferring feature importance with uncertainties in high-dimensional data

Pål Vegard Johnsen[a,b], Inga Strümke[c,d], Signe Riemer-Sørensen[a], Andrew Thomas DeWan[e], Mette Langaas[b]

[a] *SINTEF DIGITAL, 0373, Oslo, Norway*

[b] *Department of Mathematical Sciences, Norwegian University of Science and Technology, 7491, Trondheim, Norway*

[c] *Department of Engineering Cybernetics, Norwegian University of Science and Technology, 7034, Trondheim, Norway*

[d] *Department of Holistic Systems, SimulaMet, 0167, Oslo, Norway*

[e] *Department of Chronic Disease Epidemiology and Center for Perinatal, Pediatric and Environmental Epidemiology, Yale School of Public Health, CT 06510, New Haven, Connecticut, USA*

Abstract

Estimating feature importance is an essential aspect of explaining data-based models. Besides explaining the model itself, an equally relevant question is which features are important in the underlying data generating process. We present a Shapley value based framework for inferring the importance of individual features, including uncertainty in the estimator. We build upon the recently published feature importance measure of SAGE (Shapley additive global importance) and introduce sub-SAGE which can be estimated without resampling for tree-based models. We argue that the uncertainties can be estimated from bootstrapping and demonstrate the approach for tree ensemble methods. The framework is exemplified on synthetic data as well as high-dimensional genomics data.

Keywords: Interpretable machine learning, explainable artificial intelligence, Shapley values, feature importance, uncertainty estimation

1. Introduction

With the strong improvement of black-box machine learning models such as gradient boosting models and deep neural networks, the question of how to infer feature importance in these types of models has become increasingly important. The Shapley decomposition, a solution concept from cooperative game theory (Shapley, 1953), has enjoyed a surge of interest in the literature on explainable artificial intelligence in recent years, (cf. Aas et al. (2021); Lundberg et al. (2020); Sellereite and Jullum (2019); Lundberg and Lee (2017); Strumbelj and Kononenko (2013, 2010); Lundberg et al. (2019); Redelmeier et al. (2020); Kwon et al. (2021); Song et al. (2016); Moehle et al. (2021); Covert et al. (2020a); Keinan et al. (2003); Fryer et al. (2021b)). A widely used Shapley based framework for deriving feature importances in a fitted machine learning model is Shapley additive explanations (SHAP) (Lundberg and Lee, 2017; Lundberg et al., 2020), which explains single predictions' deviations from the average model prediction. As such, SHAP attributes feature importances as they are perceived by the *model*. The more recently introduced Shapley additive global importance (SAGE) is also based on the Shapley decomposition, but attributes feature importances by a global decomposition of the model loss across a whole data set (Covert et al., 2020b). The SAGE framework thus provides an explanation of the influence of the features taking into account not only the model, but also implicitly the data via the loss function, thus encapsulating that the model might not be – and most likely isn't – a perfect description of the data (see Fryer et al., 2021a, for a discussion and comparison between SHAP and SAGE as feature performance measures).

The SAGE value needs to be estimated, and the SAGE estimator is itself a random variable as the corresponding SAGE estimate is based on data of finite size generated from some unknown probability distribution. As is the case for any feature importance measure, we argue that the uncertainty in the estimate is equally important as the estimate itself for drawing conclusions. However, even computation of the SAGE-estimate is infeasible for high-dimensional data, and thus further approximations are needed (Covert et al., 2020b). To this end, we introduce sub-SAGE, which is motivated by SAGE but can be estimated exactly for tree-ensemble models, by using a reduced subset of coalitions. Additionally, we describe how to estimate a confidence interval of the sub-SAGE value. No calculation of such uncertainty exists in the SAGE package or the literature. We do this using paired bootstrapping, and demonstrate its calculation on simulated as well as observed high-dimensional data. We argue that this procedure provides a way to infer the true importance of a feature in the underlying data. We restrict ourselves to tree ensemble models. The remainder of this paper is structured as follows. In section 2 we introduce background concepts such as the Shapley value, SHAP and SAGE, before moving on to sub-SAGE in section 3 and its uncertainty in section 4. The method is exemplified in section 5 and section 6 before we discuss the results in section 7.

2. Background

In this section, we provide a brief introduction to the Shapley decomposition-based SHAP and SAGE frameworks, and how to apply these to tree ensemble models. The Shapley decomposition is a solution concept from cooperative game theory (Shapley, 1953). It provides a decomposition of *any* value function $v(\mathcal{S})$ that characterises the game, and produces a single real number, or payoff, per set of players in the game. The resulting decomposition satisfies the three properties of efficiency, monotonicity and symmetry, and is provably the only method to satisfy all three (Young, 1985; Huettner and Sunder, 2012, Thm. 2). For details see Appendix E.

Consider a supervised learning task characterised by a set of M features $\mathbf{x}_i$ and corresponding univariate[1] responses y_i, for $i = 1, \ldots, N$, and a fitted model that is a mapping from feature values to response values, i.e. $\mathbf{x}_i \rightarrow \hat{y}(\mathbf{x}_i)$. As usual, uppercase letters denote random variables while lowercase letters denote observed data values. In this work, we assume independent features, meaning $E[X_j|X_k = x_k] = E[X_j] \ \forall \ j \neq k$.

2.1. The SHAP value

Let $\mathcal{S} \subseteq \mathcal{M} \setminus \{k\}$, with $\mathcal{M} = \{1, \ldots, M\}$, denote a subset of all features not including feature k. Denote $\bar{\mathcal{S}}$ the corresponding complement subset of excluded features ($\mathcal{S} \cup \bar{\mathcal{S}} = \mathcal{M}$). The SHAP value, $\phi_k^{\mathrm{SHAP}}(\mathbf{x}, \hat{y})$, introduced

[1] The procedures described in this paper can be generalised to multivariate responses, but this renders the derivations more convoluted.

by Lundberg and Lee (2017), for a feature with index k with respect to feature values $\mathbf{x}$ and a corresponding fitted model $\hat{y}$, is defined as

$$\phi_k^{\mathrm{SHAP}}(\mathbf{x}, \hat{y}) = \sum_{\mathcal{S} \subseteq \mathcal{M} \backslash \{k\}} \frac{|\mathcal{S}|!(M - |\mathcal{S}| - 1)!}{M!} \left[v_{\mathbf{x},\hat{y}}(\mathcal{S} \cup \{k\}) - v_{\mathbf{x},\hat{y}}(\mathcal{S}) \right] . \tag{1}$$

Here, the value function $v_{\mathbf{x},\hat{y}}(\mathcal{S})$ is defined as the expected output of a prediction model conditioned that only a subset $\mathcal{S}$ of all features are included in the model,

$$v_{\mathbf{x},\hat{y}}(\mathcal{S}) = E_{\mathbf{X}_{\overline{\mathcal{S}}}}[\hat{y}(\mathbf{X}|\mathbf{X}_{\mathcal{S}} = \mathbf{x}_{\mathcal{S}})] \, . \tag{2}$$

For instance, if $\mathbf{x}_{\overline{\mathcal{S}}}$ is continuous and we assume all features to be mutually independent, we have

$$\begin{aligned} E_{\mathbf{X}_{\overline{\mathcal{S}}}}[\hat{y}(\mathbf{X}|\mathbf{X}_{\mathcal{S}} = \mathbf{x}_{\mathcal{S}})] &= \int_{\mathbf{x}_{\overline{\mathcal{S}}}} \hat{y}\left(\mathbf{X}_{\mathcal{S}} = \mathbf{x}_{\mathcal{S}}, \mathbf{X}_{\overline{\mathcal{S}}} = \mathbf{x}_{\overline{\mathcal{S}}}\right) p\left(\mathbf{X}_{\overline{\mathcal{S}}} = \mathbf{x}_{\overline{\mathcal{S}}} | \mathbf{X}_{\mathcal{S}} = \mathbf{x}_{\mathcal{S}}\right) d\mathbf{x}_{\overline{\mathcal{S}}} \\ &= \int_{\mathbf{x}_{\overline{\mathcal{S}}}} \hat{y}\left(\mathbf{X}_{\mathcal{S}} = \mathbf{x}_{\mathcal{S}}, \mathbf{X}_{\overline{\mathcal{S}}} = \mathbf{x}_{\overline{\mathcal{S}}}\right) p\left(\mathbf{X}_{\overline{\mathcal{S}}} = \mathbf{x}_{\overline{\mathcal{S}}}\right) d\mathbf{x}_{\overline{\mathcal{S}}} \, . \end{aligned} \tag{3}$$

The stochastic behaviour in $\hat{y}(\mathbf{X}|\mathbf{X}_{\mathcal{S}} = \mathbf{x}_{\mathcal{S}})$ is due to the random vector $\mathbf{X}_{\bar{\mathcal{S}}}$ of unknown feature values. We can think of the difference $v_{\mathbf{x},\hat{y}}(\mathcal{S} \cup \{k\}) - v_{\mathbf{x},\hat{y}}(\mathcal{S})$ as the mean difference in a single model prediction when using feature k in the model compared to when the value of feature k is absent. Therefore, the SHAP value can be interpreted as a feature importance measure for each single model prediction. The larger absolute SHAP value a feature k has in a single prediction, the more influence the feature is regarded to have.

2.2. The SAGE value

Define a loss function $\ell(y_i, \hat{y}(\mathbf{x}_i))$ as a measure of how well the fitted model $\hat{y}(\mathbf{x}_i)$ maps the features to a response, compared to the true response value y_i. As defined in Covert et al. (2020b), we take the SAGE value function $w(\mathcal{S})$ as the expected difference in the observed value of the loss function when the features in $\mathcal{S}$ are included in the model compared to excluding all features,

$$w_{\mathbf{X},Y,\hat{y}}(\mathcal{S}) = E_{\mathbf{X},Y}[\ell(Y, V_{\mathbf{X},\hat{y}}(\emptyset))] - E_{\mathbf{X},Y}[\ell(Y, V_{\mathbf{X},\hat{y}}(\mathcal{S}))] \, . \tag{4}$$

Here, $\emptyset$ denotes the empty set, while $V_{\mathbf{X},\hat{y}}(\mathcal{S})$ is the stochastic version of eq. (2). Specifically, $V_{\mathbf{X},\hat{y}}(\mathcal{S})$ is a random variable since its observed value varies depending on the random vector $X_{\mathcal{S}}$, while $v_{\mathbf{x},\hat{y}}(\mathcal{S})$ is a constant as we condition on the *observed* vector $\mathbf{x}_{\mathcal{S}}$. For instance, for the case where $\mathbf{x}$ and y are continuous, the expected value of the loss function when only a subset $\mathcal{S}$ of feature values are known is

$$E_{\mathbf{X},Y}[\ell(Y, V_{\mathbf{X},\hat{y}}(\mathcal{S}))] = \int_y \int_{\mathbf{x}_{\mathcal{S}}} \ell\left(y(\mathbf{x}), E_{\mathbf{X}_{\bar{\mathcal{S}}}}\left[\hat{y}\left(\mathbf{X}|\mathbf{X}_{\mathcal{S}} = \mathbf{x}_{\mathcal{S}}\right)\right]\right) p(y|\mathbf{x}_{\mathcal{S}}) p(\mathbf{x}_{\mathcal{S}}) d\mathbf{x}_{\mathcal{S}} dy \,. \quad (5)$$

Notice that the computation of $v_{\mathbf{x},\hat{y}}(\mathcal{S}) = E_{\mathbf{X}_{\bar{\mathcal{S}}}}\left[\hat{y}\left(\mathbf{X}|\mathbf{X}_{\mathcal{S}} = \mathbf{x}_{\mathcal{S}}\right)\right]$ happens inside the loss function, which is usually non-linear. Also notice that in eq. (5), we integrate over *all* possible values of $X_{\mathcal{S}}$.

The SAGE value for a feature k is defined as

$$\phi_k^{\text{SAGE}}(\mathbf{X}, Y, \hat{y}) = \sum_{\mathcal{S}\subseteq\mathcal{M}\backslash\{k\}} \frac{|\mathcal{S}|!(M - |\mathcal{S}| - 1)!}{M!} \left[w_{\mathbf{X},Y,\hat{y}}(\mathcal{S}\cup\{k\}) - w_{\mathbf{X},Y,\hat{y}}(\mathcal{S})\right] \,. \quad (6)$$

We can think of the difference $w_{\mathbf{X},Y,\hat{y}}(\mathcal{S}\cup\{k\}) - w_{\mathbf{X},Y,\hat{y}}(\mathcal{S})$ as the expected difference in the loss function when including feature k in the model compared to excluding feature k with respect to the subset $\mathcal{S}$ of known feature values. SAGE is therefore a global feature importance measure, as opposed to the SHAP value, as it does not evaluate a single prediction, but rather the impact feature k has across all predictions. The use of the loss function in the SAGE definition also makes sure that the feature importance is not only based on the model, as for the SHAP value, but also on the data itself.

The features and response can be both continuous and discrete. In the discrete case, integrals must replaced by sums and vice versa in eqs. (3) and (5). The expressions in eqs. (2) and (4) are in general unknown and need to be estimated for each choice of model and loss function. Consequently, the SHAP and SAGE values become estimates as well.

An interpretation of SAGE is that a positive SAGE value for a features implies that including this feature in the model reduces the expected model loss compared to when not including the feature.

2.3. Tree ensemble models

Consider a tree ensemble model consisting of several regression trees $f_\tau(\mathbf{x}_i)$ with predicted response $\hat{y}(\mathbf{x}_i)$, such that $\hat{y}(\mathbf{x}_i) = \sum_{\tau=1}^{T} f_\tau(\mathbf{x}_i)$ for T trees. By

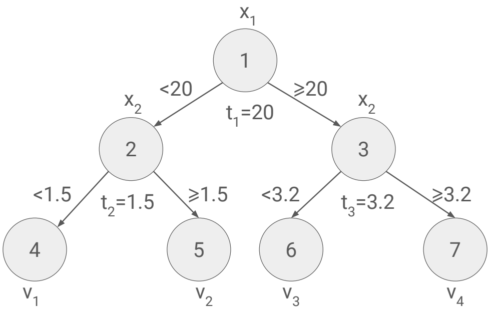


Figure 1: A regression tree including two features X_1 and X_2.

the linearity property of the expected value, we have

$$v_{\mathbf{x},\hat{y}}(\mathcal{S}) = E_{\mathbf{X}_{\bar{\mathcal{S}}}}\left[\sum_{\tau=1}^{T} f_\tau(\mathbf{X}|\mathbf{X}_\mathcal{S} = \mathbf{x}_\mathcal{S})\right] = \sum_{\tau=1}^{T} E_{\mathbf{X}_{\bar{\mathcal{S}}}}[f_\tau(\mathbf{X}|\mathbf{X}_\mathcal{S} = \mathbf{x}_\mathcal{S})]\,. \tag{7}$$

The computation of $E_{\mathbf{X}_{\bar{\mathcal{S}}}}[f_\tau(\mathbf{X}|\mathbf{X}_\mathcal{S} = \mathbf{x}_\mathcal{S})]$ can be understood through a simple example: Consider the regression tree illustrated in fig. 1. It has depth two and splits on the two features indexed 1 and 2, which are continuous and mutually independent. The regression tree has parameters such as *splitting points*, t_j, for branch nodes, and *leaf values* v_j, for leaf nodes. Assume that $x_2 = 3$ is observed. We then have

$$\begin{aligned} E_{\mathbf{X}_{\bar{\mathcal{S}}}}[f_\tau(\mathbf{X}|\mathbf{X}_\mathcal{S} = \mathbf{x}_\mathcal{S})] &= E_{X_1}[f_\tau(X_1|X_2 = 3)] \\ &= P(X_1 \geq 20)v_3 + P(X_1 < 20)v_2\,. \end{aligned} \tag{8}$$

In general, we do not know the value of $P(X_1 \leq 20)$, and need to estimate it. Consider N data instances with recorded feature values from feature k. An *unbiased* estimate of $P(X_k \leq t)$ is then

$$\hat{P}(X_k \leq t) = \frac{1}{N}\sum_{i=1}^{N} I(x_{i,k} \leq t)\,, \tag{9}$$

where $x_{i,k}$ is the observed value of feature k for data instance i. Using this estimate, we can also get an unbiased estimate for eq. (8). An unbiased estimate of $E_{\mathbf{X}_{\bar{\mathcal{S}}}}[f_\tau(\mathbf{X}|\mathbf{X}_\mathcal{S} = \mathbf{x}_\mathcal{S})]$ for any regression tree can be achieved by a recursive algorithm (Lundberg et al., 2020) with running time $O(L2^M)$, where L is the number of leaves, see algorithm 1.

Algorithm 1 Recursive algorithm for computation of $E_{\mathbf{X}_{\bar{\mathcal{S}}}}[f_\tau(\mathbf{X}|\mathbf{X}_\mathcal{S} = \mathbf{x}_\mathcal{S})]$.

```
Input: Tree $f_\tau$ with depth $d$, leaf values $\mathbf{v} = (v_1, \ldots, v_{2^d})$, feature used for splitting $\mathbf{f} = (f_1, \ldots, f_{2^d-1})$ and corresponding splitting points $\mathbf{t} = (t_1, \ldots, t_{2^d-1})$. Estimated probabilities of ending at a node $j$ given previous information, for all nodes in the tree, $\mathbf{p} = (p_1, \ldots, p_{2^d-1})$, by using some data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$ of size $N$. The subset of features $\mathcal{S}$ with corresponding known values $x_\mathcal{S}$. The left and right descendant node for each internal node $\mathbf{l} = (l_1, \ldots, l_{2^d-1})$ and $\mathbf{r} = (r_1, \ldots, r_{2^d-1})$. The index of a node $j$ in the tree $f_\tau$.
**Function** CondExpTree($j, f_\tau, \mathbf{v}, \mathbf{t}, \mathbf{f}, \mathbf{l}, \mathbf{r}, \mathbf{p}$)
**if** IsLeaf(j) **then**
    return $v_j$
**else**
    **if** $f_j \in \mathcal{S}$ **then**
        **if** $x_j \leq t_j$ **then**
            return CondExpTree($l_j, f_\tau, \mathbf{v}, \mathbf{t}, \mathbf{f}, \mathbf{l}, \mathbf{r}, \mathbf{p}$)
        **else**
            return CondExpTree($r_j, f_\tau, \mathbf{v}, \mathbf{t}, \mathbf{f}, \mathbf{l}, \mathbf{r}, \mathbf{p}$)
        **end if**
    **else**
        return CondExpTree($l_j, f_\tau, \mathbf{v}, \mathbf{t}, \mathbf{f}, \mathbf{l}, \mathbf{r}, \mathbf{p}$) $p_{l_j}$ +
                CondExpTree($r_j, f_\tau, \mathbf{v}, \mathbf{t}, \mathbf{f}, \mathbf{l}, \mathbf{r}, \mathbf{p}$) $p_{r_j}$
    **end if**
**end if**
**End Function**
CondExpTree($1, f_\tau, \mathbf{v}, \mathbf{t}, \mathbf{f}, \mathbf{l}, \mathbf{r}, \mathbf{p}$)                    ▷ Start at root node.
```

2.4. SAGE in practice

In practice, as the expressions in eq. (2) and eq. (4) must be estimated, we get a SAGE estimator rather than a SAGE value. However, since the SAGE estimator requires summing over all 2^{M-1} subsets $\mathcal{S} \subseteq \mathcal{M} \setminus \{k\}$, for *each* feature, computing the SAGE estimator for observed data with many features becomes infeasible. In Covert et al. (2020b), the SAGE estimate is approximated through a Monte Carlo simulation process. Specifically, instead of iterating over all 2^{M-1} subsets, a subset $\mathcal{S}$ is randomly sampled with replacement in each iteration out of I iterations in total. The differences $w_{\mathbf{X},Y,\hat{y}}(\mathcal{S} \cup \{k\}) - w_{\mathbf{X},Y,\hat{y}}(\mathcal{S})$ for each $\mathcal{S}$ are estimated by sampling data instances with replacement and computing sample means (see Covert et al., 2020b, Appendix D for details). For an arbitrarily large data set, the authors show convergence to the true SAGE estimate as $I \to \infty$. Among other things, both the accuracy and convergence speed of the algorithm naturally depends on the number of features in the prediction model.

Keeping in mind that the SAGE *estimator* is a random variable, we argue that its uncertainty is equally important as the estimate itself. No calculation of this inherent uncertainty exists in the SAGE package or the literature [2]. To this end, we introduce *sub-SAGE*, which is inspired by the SAGE framework, but consisting of a reduced number of subsets $\mathcal{S} \in \mathcal{Q}$. While applicable to any number of features, it is best suited for interpreting a small number of features, or a small subset of features in a large feature set.

3. Sub-SAGE

Given hundreds or thousands of features in a model, the computation time required to get a satisfactory accurate estimate of SAGE (Covert et al., 2020b), for each feature, quickly becomes unacceptable. A hybrid approach is to select a reduced subset of features of particular interest to investigate. Such a subset can for instance be selected by computing a model-based feature importance score for all features in the model and selecting the most interesting looking ones. The reduced subset of promising features can then by more thoroughly investigated in order to infer whether their model-based

[2] Covert et al. (2020b) provides the degree of convergence of the approximation of the estimate, not the uncertainty in the estimate.

importance is also reflected in the underlying data generating process. For this purpose, we introduce *sub-SAGE*, where only a selection of the in total 2^{M-1} subsets are involved in the computation of each feature.

If we want to measure the importance of a feature k based on its marginal effect, as well as potential pairwise interactions it may be involved in, computing $\mathcal{S} = \{\emptyset\}$ and $\mathcal{S} = \{m\}$ for $m = 1, \ldots, k-1, k+1, \ldots, M$ is sufficient. In addition, by including $\mathcal{S} = \{1, \ldots, k-1, k+1, \ldots, M\}$, the set of all features except feature k, this can be used to measure the importance of feature k in the presence of all features at the same time. Let $\mathcal{Q}_k$ denote the set of subsets $\mathcal{S}$ chosen above. We define the sub-SAGE value, ψ_k, for feature k as

$$\psi_k(\mathbf{X}, Y, \hat{y}) = \sum_{\mathcal{S} \in \mathcal{Q}_k} \frac{|\mathcal{S}|!(M - |\mathcal{S}| - 1)!}{3(M-1)!} \left[w_{\mathbf{X},Y,\hat{y}}(\mathcal{S} \cup \{k\}) - w_{\mathbf{X},Y,\hat{y}}(\mathcal{S}) \right] , \quad (10)$$

Each subset is weighted such that the sum of the weights of all subsets with equal size is the same for each subset size. In addition, the sum of all weights is equal to one. Hence, the construction is similar to the weights defined for Shapley values. See Appendix A for details. In this particular case, there are three possible subset sizes, and so the sum of the weights for each subset size is $\frac{1}{3}$. Shapley properties such as symmetry, dummy property and monotonicity still holds for sub-SAGE. However, as the sum is not over all possible subsets, the sub-SAGE values do no longer satisfy the efficiency axiom of the Shapley decomposition, which SHAP and SAGE do (see Appendix E) However, we regard the efficiency property as not necessary in this particular setting, as we still consider the sub-SAGE to be informative with respect to feature importance via the computed differences $w_{\mathbf{X},Y,\hat{y}}(\mathcal{S} \cup \{k\}) - w_{\mathbf{X},Y,\hat{y}}(\mathcal{S})$. In addition, the purpose is only to evaluate a small fraction of all features, not all of them. By only considering a reduced number of subsets $\mathcal{S}$, compared to SAGE, and only considering a reduced number of features to evaluate, both computing the sub-SAGE estimate as well as the uncertainty in the corresponding sub-SAGE estimator become feasible for black-box models, such as for tree ensemble models as discussed in section 4.

3.1. Using sub-SAGE to infer true relationships in the data

As the goal is to infer feature importance from a black-box model using sub-SAGE values, similar to calculating p-values without taking into account the effect of model selection, we must be extra careful. Any model selection procedure using training data is likely to overfit, resulting in a model containing false relationships that are not a general property of the population from which the data was sampled. It is therefore essential that the sub-SAGE value is calculated using independent data the model was not fitted on. We denote such independent data as test data, $(\mathbf{X}_1^0, Y_1^0), \ldots, (\mathbf{X}_{N_I}^0, Y_{N_I}^0)$, with N_I samples in total.

Consider a fitted linear regression model $\hat{y}_i = \hat{\boldsymbol{\beta}}^T \boldsymbol{x}_i$. By using test data independent of the data used for constructing the linear regression model, and using the squared error loss, one can show that for a feature k, and any $\mathcal{S} \in \mathcal{Q}_k$ (see Appendix C):

$$w_{\mathbf{X},Y,\hat{y}}(\mathcal{S} \cup \{k\}) - w_{\mathbf{X},Y,\hat{y}}(\mathcal{S}) = 2\hat{\beta}_k \operatorname{Cov}(Y, X_k) - \hat{\beta}_k^2 \operatorname{Var}(X_k). \tag{11}$$

As the expression is independent of the subset $\mathcal{S}$, this is also equal to the sub-SAGE value of feature k.

The first term in eq. (11) can be interpreted as the extent to which the influence of feature k based on the model, constructed using training data, is reflected in the independent test data. If the signs of $\hat{\beta}_k$ and $\operatorname{Cov}(Y, X_k)$ are identical, the first term is positive. If they differ, the sub-SAGE value will always be negative since the second term in eq. (11) is always negative. The second term $\hat{\beta_k}^2 \operatorname{Var}(X_k)$ is equal to the increased variance in the model by including feature k. So, if the model regards the feature as important (resulting in non-zero $\hat{\beta}_k$), while the covariance between X_k and Y from the independent test data goes in the same direction (same sign as $\hat{\beta}_k$), however small, then the benefit of including feature k in the model is smaller, the larger the variance of the feature, and at some point disadvantageous for sufficiently large variance.

3.2. sub-SAGE applied on tree ensemble models

SHAP values can be shown to be estimated efficiently for tree ensemble models, even with hundreds of thousands of features (e.g. Johnsen et al., 2021), by improving algorithm 1 to get a significantly reduced running time of $O(TLD^2)$, for T trees each of tree depth D (see Lundberg et al., 2020, for details). Unfortunately, there is no similar way to reduce the running time for estimation of SAGE values, as well as sub-SAGE values, for tree ensemble models with non-linear choices of loss functions (Lundberg et al., 2020).

We consider a tree ensemble model consisting of T trees. Consider a particular feature k to compute the sub-SAGE value as well as a subset $\mathcal{S} \in \mathcal{Q}_k$. We separate the trees in the model into two groups τ_k and the complement group $(\overline{\tau}_k)$ where τ_k is the set of trees including feature k as a splitting feature. The loss function is taken to be the squared error between the response and prediction per sample, i.e. $\ell = (y(\mathbf{x}) - \hat{y}(\mathbf{x}))^2$. Then one can show that (see Appendix B for the derivation),

$$\begin{aligned}
&w_{\mathbf{X},Y,\hat{y}}(\mathcal{S} \cup \{k\}) - w_{\mathbf{X},Y,\hat{y}}(\mathcal{S}) \\
&= E_{\mathbf{X},Y}\left[(Y(\mathbf{X}) - V_{\mathbf{X},\hat{y}}(\mathcal{S}))^2\right] - E_{\mathbf{X},Y}[(Y(\mathbf{X}) - V_{\mathbf{X},\hat{y}}(\mathcal{S} \cup \{k\}))^2] \\
&= E_{\mathbf{X},Y}\Bigg[2Y(\mathbf{X})\left(\sum_{j\in\tau_k} V_{\mathbf{X},f_j}(\mathcal{S} \cup \{k\}) - V_{\mathbf{X},f_j}(\mathcal{S})\right) + \left(\sum_{j\in\tau_k} V_{\mathbf{X},f_j}(\mathcal{S})\right)^2 \\
&- \left(\sum_{j\in\tau_k} V_{\mathbf{X},f_j}(\mathcal{S} \cup \{k\})\right)^2 + 2\left(\sum_{j\notin\tau_k} V_{\mathbf{X},f_j}(\mathcal{S})\right)\left(\sum_{j\in\tau_k} V_{\mathbf{X},f_j}(\mathcal{S} \cup \{k\}) - V_{\mathbf{X},f_j}(\mathcal{S})\right)\Bigg].
\end{aligned} \tag{12}$$

A commonly used loss function for binary classification problems is binary cross-entropy, $\ell = -y(\mathbf{x})\log\hat{y}(\mathbf{x}) - (1-y(\mathbf{x}))\log(1-\hat{y}(\mathbf{x})) = (1-y(\mathbf{x}))\sum_{j=1}^{T} f_j(\mathbf{x}) + \log\left(1 + e^{-\sum_{j=1}^{T} f_j(\mathbf{x})}\right)$. For this loss function, one can show that (see Appendix B)

$$\begin{aligned}
&w_{\mathbf{X},Y,\hat{y}}(\mathcal{S} \cup \{k\}) - w_{\mathbf{X},Y,\hat{y}}(\mathcal{S}) \\
&= E_{\mathbf{X},Y}\left[(1 - Y(\mathbf{X}))\sum_{j=1}^{T} V_{\mathbf{X},f_j}(\mathcal{S}) + \log\left(1 + \exp\left(-\sum_{j=1}^{T} V_{\mathbf{X},f_j}(\mathcal{S})\right)\right)\right] \\
&- E_{\mathbf{X},Y}\left[(1 - Y(\mathbf{X}))\sum_{j=1}^{T} V_{\mathbf{X},f_j}(\mathcal{S} \cup \{k\}) + \log\left(1 + \exp\left(-\sum_{j=1}^{T} V_{\mathbf{X},f_j}(\mathcal{S} \cup \{k\})\right)\right)\right] \\
&= E_{\mathbf{X},Y}\Bigg[(1 - Y(\mathbf{X}))\left(\sum_{j\in\tau_k} V_{\mathbf{X},f_j}(\mathcal{S}) - V_{\mathbf{X},f_j}(\mathcal{S} \cup \{k\})\right) \\
&\quad + \log\left(\frac{1 + \exp\left(-\sum_{j\in\tau_k} V_{\mathbf{X},f_j}(\mathcal{S}) - \sum_{j\notin\tau_k} V_{\mathbf{X},f_j}(\mathcal{S})\right)}{1 + \exp\left(-\sum_{j\in\tau_k} V_{\mathbf{X},f_j}(\mathcal{S} \cup \{k\}) - \sum_{j\notin\tau_k} V_{\mathbf{X},f_j}(\mathcal{S} \cup \{k\})\right)}\right)\Bigg].
\end{aligned} \tag{13}$$

3.2.1. Plug-in estimates

As discussed earlier, the expression $w_{\mathbf{X},Y,\hat{y}}(\mathcal{S} \cup \{k\}) - w_{\mathbf{X},Y,\hat{y}}(\mathcal{S})$ needs to be estimated for each $\mathcal{S} \in \mathcal{Q}_k$, *and* based on data, $(\mathbf{x}_1^0, y_1^0), \ldots, (\mathbf{x}_{N_I}^0, y_{N_I}^0)$, never used during training of the model. Let $\hat{v}_{\mathbf{x}^0,y^0,f_\tau}(\mathcal{S})$ for a particular observation $(\mathbf{x}^0, y^0)$ and regression tree f_τ denote the estimate of $v_{\mathbf{x}^0,f_\tau}(\mathcal{S}) = E_{\mathbf{X}_{\bar{\mathcal{S}}}}[f_\tau(\mathbf{X}^0|\mathbf{X}^0_{\mathcal{S}} = \mathbf{x}^0_{\mathcal{S}})]$ as described in algorithm 1. A plug-in *estimate* of ψ_k, denoted $\hat{\psi}_k$, for a regression problem with continuous response, for a tree ensemble model using the squared error loss is given by

$$\hat{\psi}_k = \sum_{\mathcal{S}\in\mathcal{Q}} \frac{|\mathcal{S}|!(M-|\mathcal{S}|-1)!}{3(M-1)!} \left[\frac{2}{N_I}\sum_{i=1}^{N_I} y_i^0 \left(\sum_{j\in\tau_k} \hat{v}_{\mathbf{x}_i^0,f_j}(\mathcal{S}\cup\{k\}) - \hat{v}_{\mathbf{x}_i^0,f_j}(\mathcal{S}) \right) \right.$$
$$+\frac{1}{N_I}\sum_{i=1}^{N_I}\left(\sum_{j\in\tau_k}\hat{v}_{\mathbf{x}_i^0,f_j}(\mathcal{S})\right)^2 - \frac{1}{N_I}\sum_{i=1}^{N_I}\left(\sum_{j\in\tau_k}\hat{v}_{\mathbf{x}_i^0,f_j}(\mathcal{S}\cup\{k\})\right)^2 \tag{14}$$
$$\left. +\frac{2}{N_I}\sum_{i=1}^{N_I}\left(\sum_{j\notin\tau_k}\hat{v}_{\mathbf{x}_i^0,f_j}(\mathcal{S})\right)\left(\sum_{j\in\tau_k}\hat{v}_{\mathbf{x}_i^0,f_j}(\mathcal{S}\cup\{k\}) - \hat{v}_{\mathbf{x}_i^0,f_j}(\mathcal{S})\right)\right].$$

The corresponding plug-in estimate for the binary cross-entropy loss given in eq. (13) can be found in a similar fashion, basically by estimating expected values as their corresponding sample means. For tree ensemble models with tree stumps (maximum depth of one for each tree), the estimate in (14) is further reduced and can be expressed as sample variance and covariance terms, see Appendix D.

4. Inference of sub-SAGE via bootstrapping

The importance of any feature may be evaluated by estimating sub-SAGE values. Similar to SAGE, a positive sub-SAGE value for a feature k indicates that including the feature in the model is expected, based on the subsets $\mathcal{S} \in \mathcal{Q}_k$, to reduce the loss function. However, the corresponding sub-SAGE plug-in *estimator* given the data generating process $(\mathbf{X}^0_1, Y^0_1), \dots, (\mathbf{X}^0_{N_I}, Y^0_{N_I})$ from some unknown probability distribution includes uncertainty, and this should be evaluated before making any assumptions about feature importance. The complexity of the sub-SAGE plug-in estimators makes paired bootstrapping a tempting approach. Specifically, the procedure is to iteratively, given independent data points at hand $(\mathbf{x}^0_1, y^0_1), \dots, (\mathbf{x}^0_{N_I}, y^0_{N_I})$, resample the data points *with replacement* to get a new bootstrapped sample $(\mathbf{x}^*_1, y^*_1), \dots, (\mathbf{x}^*_{N_I}, y^*_{N_I})$. For each bootstrapped sample, a corresponding plug-in estimate, $\hat{\psi}^*_b$, can be computed, and after B iterations, the sample $(\hat{\psi}^*_1, \dots, \hat{\psi}^*_B)$ can approximate B realizations arising from the true distribution of the plug-in estimator. A $1-2\alpha$ confidence interval can be approximated by the *percentile interval* given by $[\hat{\psi}^{*(\alpha)}, \hat{\psi}^{*(1-\alpha)}]$, where $\hat{\psi}^{*(\alpha)}$ is the 100α empirical percentile, meaning the $B \cdot \alpha$th least value in the ordered list of the samples $(\hat{\psi}^*_1, \ldots, \hat{\psi}^*_B)$[3]. The accuracy in the percentile interval increases for larger number of bootstrap iterations. A typical number is $B = 1000$ regarded to be sufficient in most cases. The algorithm of the paired bootstrap applied specifically to tree ensemble models is given in algorithm 2. Notice that for each bootstrap sample, the probability estimates in the trees need to be updated according to eq. (9). In situations where the plug-in estimator is biased, or there is skewness in the corresponding distribution, the bias-corrected and accelerated bootstrap, first introduced in Efron (1987), may give even more accurate confidence intervals at the cost of considerable increase in computational efforts.

Algorithm 2 Paired bootstrap of sub-SAGE value with percentile interval

```
Given independent test data $(\mathbf{x}^0_1, y^0_1), \ldots, (\mathbf{x}^0_{N_I}, y^0_{N_I})$, model $\hat{y}(\mathbf{x}) = \sum_{\tau=1}^{T} f_\tau(\mathbf{x})$, feature $k$, a loss function and $\alpha$ to estimate $1 - 2\alpha$ confidence interval:
Preallocate vector BootVec of length $B$, the total number of bootstrap iterations.
for $b = 1, 2, \ldots, B$ do
    Resample data $N_I$ times with replacement to get
    $(\mathbf{x}^*_1, y^*_1), \ldots, (\mathbf{x}^*_{N_I}, y^*_{N_I})$
    Update probabilities estimates in all the trees in $\hat{y}(\mathbf{x})$ to get $\mathbf{p}^*$
    BootVec[b] = $\hat{\psi}^*_k$
end for
Percentile interval given by $[\hat{\psi}^{*(\alpha)}, \hat{\psi}^{*(1-\alpha)}]$
```

5. Proof of concept - With known underlying data generating process

In this section, we exemplify the sub-SAGE method on synthetic data with a known relationship defined as

$$\begin{aligned} f(\mathbf{X}_i) = a_0 + a_1 X_{i,1} + a_2 X_{i,2} + a_{21} X_{i,1} e^{X_{i,2}} + a_3 X_{i,3}^2 + a_4 \sin(X_{i,4}) \\ a_5 \log(1 + X_{i,5}) - X_{i,5} I(X_{i,6} > 7) + \epsilon_i \,, \end{aligned} \tag{15}$$

[3] Assuming $B \cdot \alpha$ is an integer. See for instance Efron and Tibshirani (1994) for conventions.

with $a_0 = -0.5, a_1 = 0.03, a_2 = -0.05, a_{21} = 0.3, a_3 = 0.02, a_4 = 0.35, a_5 = -0.2$, and where the features are sampled from the following distributions

$$\begin{aligned} X_1 &\sim \text{Binom}(\text{size} = 2, p = 0.4) \\ X_2 &\sim \text{Binom}(\text{size} = 2, p = 0.04) \\ X_3 &\sim \Gamma(\text{shape} = 10, \text{rate} = 2) \\ X_4 &\sim \text{Unif}(0, \pi) \\ X_5 &\sim \text{Poisson}(\lambda = 15) \\ X_6 &\sim \text{N}(\mu = 0, \sigma = 10) \\ \epsilon_i &\sim \text{N}(\mu = 0, \sigma = 2)\,. \end{aligned} \tag{16}$$

In addition, we generate 94 noise variables. $j = 7, \ldots, 47$ with a normal distribution $X_j \sim \text{N}(\mu_j, \sigma_j)$ and $j = 48, \ldots, 100$ with a binomial distribution $X_j \sim \text{Binom}(2, p_j)$ where μ_j, σ_j and p_j are sampled from a uniform distribution. Data is generated to give a total of 16000 samples, and then separated randomly in three disjoint subsets: Data for training (50%), data for evaluation during training (30%) and independent test data (20%) used for estimating sub-SAGE values. We fit an ensemble tree model using XGBoost (Chen and Guestrin, 2016) to the true influential features $1, \ldots, 6$ together with the noise variables $7, \ldots, 100$.

The hyperparameters are fixed to max_depth = 2, learning rate $\eta = 0.05$, subsample = 0.7, regularization parameters $\lambda = 1$, $\gamma = 0$ and colsample_bytree = 0.8 with early_stopping_rounds = 20 using training data ($n = 8000$) and validation data ($n = 4800$). See (Chen and Guestrin, 2016) for details about the hyperparameters. We apply the squared error loss during training. This results in a final model including a total of 230 trees and 62 unique features out of the 100 input-features.

From the trained model, each feature is given a score to evaluate its feature importance *based on the model.* We apply the expected relative feature contribution (ERFC), given N data points, introduced in Johnsen et al. (2021), which is basically a summary score from the corresponding SHAP values for each feature and individual data point,

$$\kappa_k = \sum_{i=1}^{N} \frac{|\phi_{i,k}^{\text{SHAP}}(\mathbf{x}_i, \hat{y})|}{|\phi_0^{\text{SHAP}}| + \sum_{j=1}^{K} |\phi_{i,j}^{\text{SHAP}}(\mathbf{x}_i, \hat{y})|}\,, \tag{17}$$

with $\phi_0^{\text{SHAP}} = v_{\mathbf{x},\hat{y}}(\emptyset)$. The ERFCs scores can be computed based on the data used to construct the model, as we only need to measure what the model considers important. The features with the largest ERFC-values are then considered the most promising ones *based on the model.* Depending on your hypothesis of interest, one can evaluate the uncertainty in the feature importance by computing sub-SAGE

Table 1: The resulting ranking based on the expected relative feature contribution (ERFC) after having trained an XGBoost model consisting of 6 influential features and 94 noise features.

Feature	**ERFC**
x_6	0.48
x_5	0.060
x_3	0.026
x_1	0.022
x_4	0.0036
x_2	0.0030
x_{12}	0.0028
x_{30}	0.0022
x_{40}	0.0019

estimates with corresponding bootstrap-derived percentile intervals. However, it is important that the sub-SAGE estimates are calculated based on independent test data never used during training. From the trained model, we compute the ERFC based on the training data and validation data together ($n = 12800$), and table 1 shows the top 10 features with the largest ERFC-values. This shows that the XGBoost model has accurately ranked the most influential features among the top 10 list, for this rather simple relationship. These scores, based on SHAP values, are only with respect to what the *model* considers important. The sub-SAGE can now be applied to infer whether the importance of any feature from the model is also reflected in the data. As an example, let us consider features 6, 1, 2 and 12 where feature 6 has a strong influence, feature 1 has a weaker influence, and feature 2 has the weakest influence, while feature 12 has no influence with respect to $f(\mathbf{x}_i)$ in eq. (15). Their sub-SAGE estimate along with histograms to estimate the corresponding distribution of the sub-SAGE estimators are shown in fig. 2 for training plus validation data as well as for independent test data. We see that sub-SAGE values inferred using training data overestimates the false influence of feature 12, while using the test data correctly indicates that feature 12 has a weak or no influence. We also see from the other histograms that using the training data underestimates the uncertainty in the sub-SAGE estimate.

By using the test data for computation of the sub-SAGE estimates, the estimated 95% percentile intervals of the sub-SAGE values for each feature are 6 : (39.45, 44.15), 1 : (−0.038, 0.14), 2 : (−0.043, 0.040) and 12 : (−0.030, 0.0050). These ranges allow us to conclude that feature 6, correctly, is highly influential, while feature 12 is highly unlikely to have any influence. The added benefit of the estimated confidence intervals is to prevent us from concluding that features 1 and 2 are influential but rather concluding that feature 1 is highly likely to be

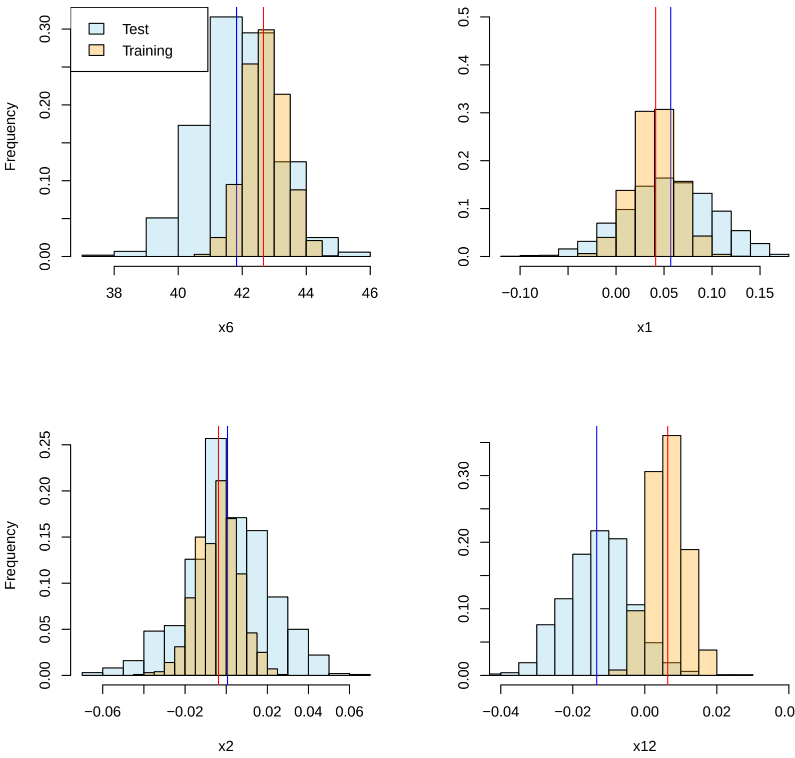


Figure 2: The estimate of the sub-SAGE, and the corresponding bootstrap distribution for the synthetic data for features x_6, x_2, x_1 and x_{12}, when applying data used during training (orange), and independent test data (blue).

influential, as its average is above zero.

To correct for a potential bias in the plug-in estimator of the sub-SAGE as well as potential changes in the standard deviation of the estimator at different levels, the bias-corrected and accelerated bootstrap confidence interval may give more accurate bootstrap confidence intervals (Efron, 1987). This results in the following intervals $6 : (39.45,\ 44.13)$, $1 : (-0.034,\ 0.14)$, $2 : (-0.047,\ 0.037)$ and $12 : (-0.031,\ 0.0040)$, with only negligible changes from the percentile confidence intervals. The sub-SAGE underestimation of the influence of both features 1 and 2, but particularly feature 2, can be explained by looking at fig. 3.

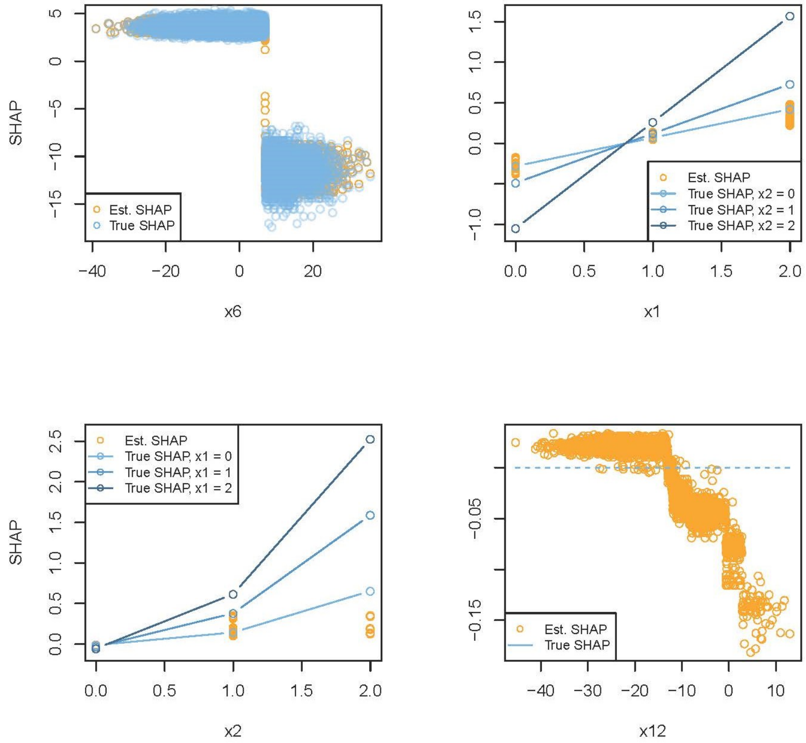


Figure 3: Comparison of true SHAP value for each data point with the estimated SHAP value from the model fitted on the synthetic data, eq. (15). The deviations explain the reasons behind under- and overestimation of feature importance.

As the data generating process is known, we can compare the true SHAP value at each point with the corresponding SHAP value from the fitted model. It shows that the influence of feature 6 is quite accurately modelled, while the effect of feature 1 and particularly feature 2 is highly underestimated when $x_1 = 1$ and $x_2 = 2$. As features 1 and 2 interact, the SHAP value of feature 1 depends on the value of feature 2. It also becomes clear that feature 12, according to the model, has a negative trend in the SHAP value, but the true SHAP value is equal to zero (no importance), regardless of the value of feature 12. See Appendix F for derivations.

6. Application on genetic data using the UK Biobank resource

To demonstrate the ability of sub-SAGE on observed data, we consider a realistic high-dimensional machine learning problem that often occurs when using genetic data, namely the influence of specific features on a given trait.

We use both genetic and non-genetic data from UK Biobank, a large prospective cohort study in the United Kingdom that began in 2006 consisting of about $500'000$ participants (Sudlow et al., 2015; Bycroft et al., 2018), and attempt to infer the influence of specific features with respect to obesity (BMI ≥ 30), by training an XGBoost model and computing sub-SAGE values.

We treat this as a classification problem between the categories obese and non-obese (see Johnsen et al., 2021, for details). Of particular interest is whether any genetic markers are important. The most used method in this setting is a so-called genome-wide association study (GWAS), where each genetic variant is tested individually in a general linear (mixed-effects) regression model (Visscher et al., 2017; Zhou et al., 2018). A corresponding p-value less than 5×10^{-8} is often considered statistically significant, a tiny significance level due to the multiple comparison problem (Goeman and Solari, 2014). When the same association is replicated in an independent data set, the association is considered to be robust.

We study the XGBoost model constructed in Johnsen et al. (2021) based on 3000 features both genetic (single nucleotide polymorphism (SNP)) and non-genetic, for $64'000$ unrelated White-British participants from UK Biobank. The genetic data consists of so-called *minor allele counts* or genotype values from SNPs (see e.g. Visscher et al., 2017) filtered to ensure independence without significant loss of information (Johnsen et al., 2021). Non-genetic features included are sex, age, physical activity frequency, intake of saturated fate, sleep duration, stress and alcohol consumption (see Johnsen et al., 2021, for definitions). The model is trained with hyperparameters: learning rate $\eta = 0.05$, *colsample* $=$ *subsample* $=$

Table 2: The resulting ranking based on the expected relative feature contribution (ERFC) after having trained an XGBoost model consisting of 3000 features and $64'000$ individuals from UK Biobank.

Feature	**ERFC**
Alcohol intake frequency	0.088
Genetic sex	0.086
Physical activity frequency	0.073
Intake of saturated fat	0.044
Sleep duration	0.036
Stress	0.034
Age at recruitment	0.033
rs17817449	0.017
rs489693	0.012
rs1488830	0.011
rs13393304	0.010
rs10913469	0.01
rs2820312	0.0086

$colsample\_by\_tree = 0.8$, $max\_depth = 2$, $\lambda = 1$, $\gamma = 1$, $early\_stopping\_rounds = 20$, and binary cross-entropy loss. The trained model included only 532 features among the 3000 input features spread along a total of 607 trees. The features with the largest ERFC-scores, based on the training data, and therefore considered the most promising features, are given in table 2.

While the non-genetic features are considered the most important, the most important SNP according to the model is rs17817449, a SNP connected to the FTO gene at chromosome 16, previously associated (statistically significant) with obesity in a large number of genome-wide association studies including different independent data sets (Locke et al., 2015). The SNP rs13393304 at chromosome 2 has previously been associated with obesity using UK Biobank data (Karlsson et al., 2019). The SNP rs2820312 has not previously been associated with obesity, but with hypertension based on UK Biobank data (Gagliano Taliun et al., 2020). The SNPs mentioned above are explored further by computing sub-SAGE estimates

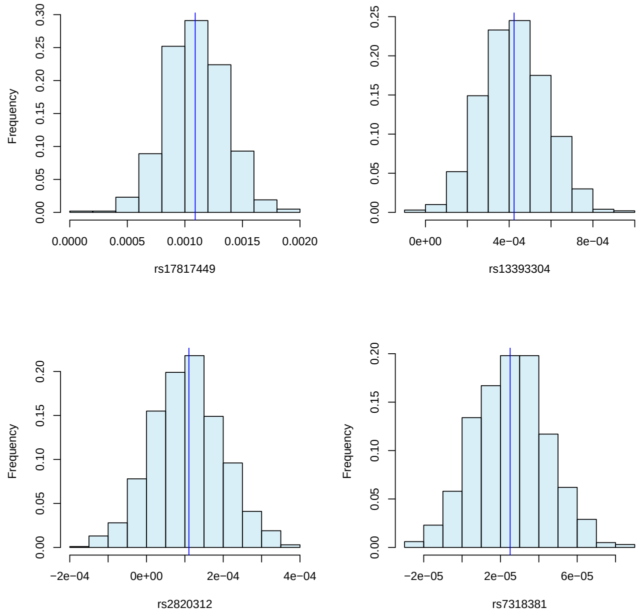


Figure 4: The estimates and corresponding uncertainties in the sub-SAGE values for the four SNPs agree with previous studies (GWAS) regarding SNP-association with obesity.

including paired bootstrap-derived percentile intervals by using $20'000$ (unrelated White-British) participants from UK Biobank not used while training the model. We also compute sub-SAGE for the randomly selected SNP rs7318381, which has never been associated with obesity, and with a small ERFCs in the XGBoost model (0.0016). The result is given in fig. 4.

The sub-SAGE values do indicate that both rs17817449 and rs13393304 are highly likely to be associated with obesity. The 95% percentile interval of the sub-SAGE value for rs17817449 is $(0.0006,\ 0.0016)$, and $(0.00014,\ 0.00073)$ for rs13393304. The SNPs rs2820312 and rs7318381 are less likely to be associated with obesity, and if they are true associations, the uncertainties in the estimates indicate that the effects are microscopic. The 95% percentile intervals for rs2820312 is $(-7.08 \cdot 10^{-5},\ 2.95 \cdot 10^{-4})$, and $(-1.13 \cdot 10^{-5},\ 6.32 \cdot 10^{-5})$ for rs7318381.

When dealing with relatively large data sizes such as for the genetic example above, the bias-corrected and accelerated bootstrap interval can become infeasible due to the estimation of the acceleration parameter. However, as the acceleration parameter is proportional to the skewness of the bootstrap distribution, and if the bootstrap distribution indeed has a small skewness, as is the case here, it is often sufficient to set the acceleration parameter equal to zero. This gives no change in the percentile intervals of rs17817449 and rs13393304, but the bias-corrected 95% bootstrap intervals of rs2820312 and rs7318381 become $(-6.10\cdot 10^{-5},\ 0.00030)$ and $(-1.18\cdot 10^{-5},\ 6.19\cdot 10^{-5})$ respectively. These are negligible changes, indicating that the plug-in estimates are low-biased.

7. Discussion and conclusion

We present a Shapley value based framework for inferring the importance of individual features, including uncertainty in the estimator. We demonstrate how to infer feature influence for a tree ensemble model with high-dimensional data using sub-SAGE and paired bootstrapping. As an example, we use XGBoost, a gradient tree-boosting model, applied to both a known data generating process, as well as realistic high-dimensional data. We emphasize the importance of using test data, independent of data used to construct the model, to compute sub-SAGE estimates.

It is important to notice that the percentile intervals, constructed to evaluate the uncertainty in the sub-SAGE estimate, themselves include uncertainty. The uncertainty of the percentile intervals depends on the number of bootstraps, B, as well as the size n of data. However, in addition, the uncertainty also depends on the ratio p/n, where p is the total number of features *used* in the model (not necessarily the number of input-features for constructing the model). This fact is particularly important in high-dimensional problems, and it has been discussed for instance in Karoui and Purdom (2018). When applied to linear models, one observation from a simulation is for instance that the paired bootstrap becomes more conservative (loss of power) the larger the ratio p/n is. Observe that for the simulation example above, $p/n = 62/3200 = 0.019$, while for the genetic data, the ratio is $p/n = 533/20000 = 0.027$, deliberately chosen to be small in order to account for the problems arising when p/n becomes too large. For the genetic data, a filtering process is first needed as the data from UK Biobank originally includes around $530'000$ SNPs and $207'000$ individuals ($p/n = 2.56$). The applied filtering method and potential pitfalls are described in Johnsen et al. (2021).

It seems reasonable to apply the same loss function in the sub-SAGE estimate as the loss function that was used to construct the model. However, there may be situations where it is meaningful to compute the sub-SAGE values for a different loss function than the loss function used during training in order to make more objective interpretations. This may e.g. be the case when the model is provided 'as is' and you do not know the training loss function, or when using adapted loss functions, e.g. weighted binary cross-entropy, but the interpretation is relevant for a standard cross-entropy.

In this work we have assumed all features to be mutually statistically independent, an unrealistic scenario in most cases, except for situations such as with genetic data where one can make sure that the genetic distance between the SNPs is sufficiently large to minimize the correlation. If many features are statistically dependent, one is required to estimate conditional expected values (see e.g. Aas et al., 2021). In a high-dimensional setting, this often becomes very tedious and even infeasible in most cases. An important line of future research to allow for easy evaluation of feature influence in a high-dimensional setting, is dimensionality reduction of the features with reduced loss of interpretation of the cluster variables created.

8. Acknowledgements

This research was supported by the Norwegian Research Council grant 272402 (PhD Scholarships at SINTEF), project number 304843 (the EXAIGON project), as well the funding for research stays abroad for doctoral and postdoctoral fellows financed by the Norwegian Research Council. The research has been conducted using the UK Biobank Resource under Application Number 32285. We thank the Yale Center for Research Computing for guidance and use of the research computing infrastructure. We thank The Gemini Center for Sepsis Research for establishing cooperation with Yale School of Public Health.

9. Code availability

Source code is available at `https://github.com/palVJ/subSAGE`.

Appendix A. The weights in sub-SAGE

The sub-SAGE, ψ_k, is defined as eq. (10) and repeated here for convenience,

$$\psi_k(\mathbf{X}, Y, \hat{y}) = \sum_{\mathcal{S} \in \mathcal{Q}_k} \frac{|\mathcal{S}|!(M - |\mathcal{S}| - 1)!}{3(M-1)!} \left[w_{\mathbf{X},Y,\hat{y}}(\mathcal{S} \cup \{k\}) - w_{\mathbf{X},Y,\hat{y}}(\mathcal{S}) \right] , \tag{A.1}$$

with $\mathcal{Q}_k$ consisting of the subsets $\{\emptyset\}$, $\{m\}$ for $m = 1, \ldots, k-1, k+1, \ldots, M$ and $\{1, 2, \ldots, k-1, k+1, \ldots, M\}$. In other words, there are three different achievable subset sizes, namely of size zero, one and $M-1$. As we want the sum of all weights to be equal to one, and that the sum of the weights of equal subset size is the same for all subset sizes, we need the corresponding weight for $\mathcal{S}=\{\emptyset\}$ and $\mathcal{S}=\{1,2,\ldots,k-1,k+1,\ldots,M\}$ to be 1/3, while the sum of the weights for $\mathcal{S}=\{m\}$ for $m=1,\ldots,k-1,k+1,\ldots,M$ needs to be 1/3. For $\mathcal{S}=\{\emptyset\}$, we see that the weight is $0!(M-1)!/3(M-1)!=1/3$ and for $\mathcal{S}=\{1,2,\ldots,k-1,k+1,\ldots,M\}$ the weight is $(M-1)!0!/3(M-1)!=1/3$, just as we wanted. For the subsets of size one, the weight is $1!(M-2)!/3(M-1)!=1/3(M-1)$. There are $M-1$ subsets of size one in total, and so the sum of the weights are also 1/3. In other words, the definition of the weights in sub-SAGE makes sure that the sum of all weights is equal to one, and that the sum of the weights of equal subset size is the same for all subset sizes.

Appendix B. Derivation of Sub-SAGE for squared error and binary cross-entropy

Using as loss function the squared error loss, the loss per sample is $\ell=(y-\hat{y})^2$. Considering a feature k for which to compute the sub-SAGE value, we separate the trees in our ensemble model into two groups: τ_k, being the set of trees including feature k as a splitting point, and its complement group $(\bar{\tau}_k)$. Then, for any $\mathcal{S}\in\mathcal{Q}_k$,

$$\begin{aligned}
&w_{\mathbf{X},Y,\hat{y}}(\mathcal{S}\cup\{k\})-w_{\mathbf{X},Y,\hat{y}}(\mathcal{S})\\
&=E_{\mathbf{X},Y}\left[\left(Y\left(\mathbf{X}\right)-V_{\mathbf{X},\hat{y}}\left(\mathcal{S}\right)\right)^2\right]-E_{\mathbf{X},Y}[\left(Y\left(\mathbf{X}\right)-V_{\mathbf{X},\hat{y}}(\mathcal{S}\cup\{k\})\right)^2]\\
&=E_{\mathbf{X},Y}\left[\left(Y-\sum_{j\in\tau_k}V_{\mathbf{X},f_j}\left(\mathcal{S}\right)-\sum_{j\notin\tau_k}V_{\mathbf{X},f_j}\left(\mathcal{S}\right)\right)^2-\left(Y-\sum_{j\in\tau_k}V_{\mathbf{X},f_j}(\mathcal{S}\cup\{k\})-\sum_{j\notin\tau_k}V_{\mathbf{X},f_j}(\mathcal{S}\cup\{k\})\right)^2\right]\\
&=E_{\mathbf{X},Y}\left[\left(Y-\sum_{j\in\tau_k}V_{\mathbf{X},f_j}\left(\mathcal{S}\right)-\sum_{j\notin\tau_k}V_{\mathbf{X},f_j}\left(\mathcal{S}\right)\right)^2-\left(Y-\sum_{j\in\tau_k}V_{\mathbf{X},f_j}(\mathcal{S}\cup\{k\})-\sum_{j\notin\tau_k}V_{\mathbf{X},f_j}\left(\mathcal{S}\right)\right)^2\right]\\
&=E_{\mathbf{X},Y}\left[2Y\left(\sum_{j\in\tau_k}V_{\mathbf{X},f_j}(\mathcal{S}\cup\{k\})-V_{\mathbf{X},f_j}\left(\mathcal{S}\right)\right)+\left(\sum_{j\in\tau_k}V_{\mathbf{X},f_j}\left(\mathcal{S}\right)\right)^2-\left(\sum_{j\in\tau_k}V_{\mathbf{X},f_j}(\mathcal{S}\cup\{k\})\right)^2\right.\\
&\left.+\,2\left(\sum_{j\notin\tau_k}V_{\mathbf{X},f_j}\left(\mathcal{S}\right)\right)\left(\sum_{j\in\tau_k}V_{\mathbf{X},f_j}(\mathcal{S}\cup\{k\})-V_{\mathbf{X},f_j}\left(\mathcal{S}\right)\right)\right],
\end{aligned}\tag{B.1}$$

having used that the two random variables $V_{\mathbf{X},f_j}(\mathcal{S}\cup\{k\})$ and $V_{\mathbf{X},f_j}\left(\mathcal{S}\right)$ are equivalent, or equal in distribution, for $j\notin\tau_k$. Note that the corresponding observed value $v_{\mathbf{x},f_j}(\mathcal{S}\cup\{k\})=E_{\mathbf{X}_{\overline{\mathcal{S}}}}[f_j(\mathbf{X}|\mathbf{X}_{\mathcal{S}}=\mathbf{x}_{\mathcal{S}\cup\{k\}})]=E_{\mathbf{X}_{\overline{\mathcal{S}}}}[f_j(\mathbf{X}|\mathbf{X}_{\mathcal{S}}=\mathbf{x}_{\mathcal{S}})]=$

$v_{\mathbf{x},f_j}(\mathcal{S})$ for all $\mathcal{S} \in \mathcal{Q}_k$ since the regression tree f_j does not include feature k, and the features are assumed mutually independent.

Using as loss function the binary cross-entropy, the loss function per sample is $\ell = -y \log \hat{y} - (1-y)\log(1-\hat{y}) = (1-y)\sum_{\tau=1}^{T} f_\tau + \log\left(1 + e^{-\sum_{\tau=1}^{T} f_\tau}\right)$. Then, we have

$$\begin{aligned}
& w_{\mathbf{X},Y,\hat{y}}(\mathcal{S}\cup\{k\}) - w_{\mathbf{X},Y,\hat{y}}(\mathcal{S}) \\
&= E_{\mathbf{X},Y}\left[(1-Y(\mathbf{X}))\sum_{\tau=1}^{T} V_{\mathbf{X},f_\tau}(\mathcal{S}) + \log\left(1+\exp\left(-\sum_{\tau=1}^{T} V_{\mathbf{X},f_\tau}(\mathcal{S})\right)\right)\right] \\
&- E_{\mathbf{X},Y}\left[(1-Y(\mathbf{X}))\sum_{\tau=1}^{T} V_{\mathbf{X},f_\tau}(\mathcal{S}\cup\{k\}) + \log\left(1+\exp\left(-\sum_{\tau=1}^{T} V_{\mathbf{X},f_\tau}(\mathcal{S}\cup\{k\})\right)\right)\right] \\
&= E_{\mathbf{X},Y}\left[(1-Y(\mathbf{X}))\left(\sum_{j\in\tau_k} V_{\mathbf{X},f_j}(\mathcal{S}) + \sum_{j\notin\tau_k} V_{\mathbf{X},f_j}(\mathcal{S})\right)\right] \\
&+ E_{\mathbf{X},Y}\left[\log\left(1+\exp\left(-\sum_{j\in\tau_k} V_{\mathbf{X},f_j}(\mathcal{S}) - \sum_{j\notin\tau_k} V_{\mathbf{X},f_j}(\mathcal{S})\right)\right)\right] \\
&- E_{\mathbf{X},Y}\left[(1-Y(\mathbf{X}))\left(\sum_{j\in\tau_k} V_{\mathbf{X},f_j}(\mathcal{S}\cup\{k\}) + \sum_{j\notin\tau_k} V_{\mathbf{X},f_j}(\mathcal{S}\cup\{k\})\right)\right] \\
&- E_{\mathbf{X},Y}\left[\log\left(1+\exp\left(-\sum_{j\in\tau_k} V_{\mathbf{X},f_j}(\mathcal{S}\cup\{k\}) - \sum_{j\notin\tau_k} V_{\mathbf{X},f_j}(\mathcal{S}\cup\{k\})\right)\right)\right] \\
&= E_{\mathbf{X},Y}\left[(1-Y(\mathbf{X}))\left(\sum_{j\in\tau_k} V_{\mathbf{X},f_j}(\mathcal{S}) - V_{\mathbf{X},f_j}(\mathcal{S}\cup\{k\})\right)\right. \\
&\quad \left. + \log\left(\frac{1+\exp\left(-\sum_{j\in\tau_k} V_{\mathbf{X},f_j}(\mathcal{S}) - \sum_{j\notin\tau_k} V_{\mathbf{X},f_j}(\mathcal{S})\right)}{1+\exp\left(-\sum_{j\in\tau_k} V_{\mathbf{X},f_j}(\mathcal{S}\cup\{k\}) - \sum_{j\notin\tau} V_{\mathbf{X},f_j}(\mathcal{S}\cup\{k\})\right)}\right)\right].
\end{aligned} \tag{B.2}$$

Appendix C. (Sub-)SAGE with multiple linear regression

Consider a fitted linear regression model $\hat{y}_i = \hat{\boldsymbol{\beta}}^T \mathbf{x}_i$, with uncorrelated features. By applying the squared error loss, and by considering $\hat{\boldsymbol{\beta}}$ as a constant (by using data not used to estimate $\hat{\boldsymbol{\beta}}$), we have for a feature k, and a subset $\mathcal{S} \in \mathcal{Q}_k$ that

$$\begin{aligned}
&w_{\mathbf{X},Y,\hat{y}}(\mathcal{S} \cup \{k\}) - w_{\mathbf{X},Y,\hat{y}}(\mathcal{S}) = E_{\boldsymbol{X},Y}[(Y - V_{\boldsymbol{X},\hat{y}}(\mathcal{S}))^2] - E_{\boldsymbol{X},Y}[(Y - V_{\boldsymbol{X},\hat{y}}(\mathcal{S} \cup \{k\}))^2] \\
&= E_{\mathbf{X},Y}[2Y\hat{\beta}_k(X_k - E[X_k]) + V_{\boldsymbol{X},\hat{y}}(\mathcal{S})^2 - V_{\boldsymbol{X},\hat{y}}(\mathcal{S} \cup \{k\})^2] \\
&= 2\hat{\beta}_k E_{\mathbf{X},Y}\left[Y(X_k - E[X_k])\right] + 2E_{\mathbf{X},Y}\left[\hat{\beta}_k(\hat{\beta}_S^T X_S + \hat{\beta}^T_{\overline{\mathcal{S}\cup\{k\}}} X_{\overline{\mathcal{S}\cup\{k\}}}])(E[X_k] - X_k)\right] \\
&- \hat{\beta}_k^2 E_{\mathbf{X},Y}\left[\left({X_k}^2 - E[X_k]^2\right)\right] \\
&= 2\hat{\beta}_k \operatorname{Cov}(Y, X_k) - \hat{\beta}_k^2 \operatorname{Var}(X_k),
\end{aligned} \tag{C.1}$$

with

$$V_{\boldsymbol{X},\hat{y}}(\mathcal{S}) = \hat{\beta}_k E[X_k] + \hat{\beta}_{\mathcal{S}} X_s + \hat{\beta}_{\overline{\mathcal{S}\cup\{k\}}} E[X_{\overline{\mathcal{S}\cup\{k\}}}],$$

the stochastic version of $v_{\boldsymbol{x},\hat{y}}(\mathcal{S}) = E[\hat{y}(\boldsymbol{X})|X_{\mathcal{S}} = x_{\mathcal{S}}] = \hat{\beta}_k E[X_k] + \hat{\beta}_{\mathcal{S}} x_s + \hat{\beta}_{\overline{\mathcal{S}\cup\{k\}}} E[X_{\overline{\mathcal{S}\cup\{k\}}}]$, and

$$V_{\boldsymbol{X},\hat{y}}(\mathcal{S} \cup \{k\}) = \hat{\beta}_k X_k + \hat{\beta}_{\mathcal{S}} X_s + \hat{\beta}_{\overline{\mathcal{S}\cup\{k\}}} E[X_{\overline{\mathcal{S}\cup\{k\}}}],$$

the stochastic version of $v_{\boldsymbol{x},\hat{y}}(\mathcal{S} \cup \{k\})$. See Appendix B in Aas et al. (2021) for derivation of $v_{\boldsymbol{x},\hat{y}}(\mathcal{S})$ in linear regression. The second term in the third line of eq. (C.1) is equal to zero since the features are independent, and $\hat{\boldsymbol{\beta}}$ is considered a constant. Notice therefore that the sub-SAGE value, as well as the SAGE-value, is independent of the subset $\mathcal{S}$ used, and equal to eq. (C.1).

The second term ${\hat{\beta}_k}^2 \operatorname{Var}(X_k)$ is in fact equal to the increased variance in the model by including feature k actively in the model since

$$\begin{aligned}
&E[V_{\boldsymbol{X},\hat{y}}(\mathcal{S})^2 - V_{\boldsymbol{X},\hat{y}}(\mathcal{S} \cup \{k\})^2] \\
&= E[V_{\boldsymbol{X},\hat{y}}(\mathcal{S})^2] - E[V_{\boldsymbol{X},\hat{y}}(\mathcal{S})]^2 - (E[V_{\boldsymbol{X},\hat{y}}(\mathcal{S} \cup \{k\})^2] - E[V_{\boldsymbol{X},\hat{y}}(\mathcal{S} \cup \{k\})]^2) \\
&= \operatorname{Var}(V_{\boldsymbol{X},\hat{y}}(\mathcal{S})) - \operatorname{Var}(V_{\boldsymbol{X},\hat{y}}(\mathcal{S} \cup \{k\})),
\end{aligned} \tag{C.2}$$

because $E[V_{\boldsymbol{X},\hat{y}}(\mathcal{S})] = E[V_{\boldsymbol{X},\hat{y}}(\mathcal{S} \cup \{k\})]$.

For linear regression models, this shows that the sub-SAGE value is only positive if the agreement between the model and the independent test data (first term in eq. (C.1)) upweights the increased variance in the model (second term in eq. (C.1)) by including feature k.

We neither know the variance of X_k nor the correlation between X_k and Y, and so these must also be estimated from the data. The sample mean and sample covariance are unbiased and consistent estimators. Therefore, by using *independent* test data $(\mathbf{x}_1^0, y_1^0), ..., (\mathbf{x}_{N_I}^0, y_{N_I}^0)$ of size N_I, the estimator of $\hat{\beta}_k$, denote it $T(\hat{\beta}_k)$, is statistically independent from the test data, and by applying the sample mean and covariance we get the following unbiased estimate of eq. (C.1)

$$\begin{aligned}
&\hat{w}_{\mathbf{X},Y,\hat{y}}(\mathcal{S} \cup \{k\}) - \hat{w}_{\mathbf{X},Y,\hat{y}}(\mathcal{S}) \\
&= \frac{2\hat{\beta}_j}{n_I - 1} \sum_{i=1}^{N_I} \left[y_i^0 x_{i,j}^0 - \left(\frac{1}{N_I} \sum_{i=1}^{N_I} x_{i,j} \right) \left(\frac{1}{N_I} \sum_{i=1}^{N_I} y_{i,j} \right) \right] - \hat{\beta_j}^2 \frac{1}{N_I - 1} \sum_{i=1}^{N_I} \left(x_{i,j} - \frac{1}{N_I} \sum_{i=1}^{N_I} x_{i,j} \right)^2 \\
&= 2\hat{\beta}_j \widehat{\mathrm{Cov}}^0 (Y, X_k) - \hat{\beta_k}^2 \widehat{\mathrm{Var}}^0 (X_k) \, .
\end{aligned} \tag{C.3}$$

If we did not use training data separately for constructing the model, and test data to compute sub-SAGE values, the second term in the third line of eq. (C.1) would no longer become zero since the estimator $T(\hat{\boldsymbol{\beta}})$ naturally is correlated with the training data itself. It may seem confusing to treat $\hat{\beta}_k$ in eq. (C.1) as a constant when the corresponding estimator $T(\hat{\beta}_k)$ indeed has a distribution based on the training data. However, one may look at the procedure of sub-SAGE as objectively observing the properties of the raw model itself without taking into account the data used for training the model.

Appendix D. Sub-SAGE estimate for tree ensemble models with tree stumps

Consider a tree ensemble model with regression trees of depth one, so-called tree stumps. Each tree stump includes exactly one feature from the set $\mathcal{M}$ of all M features. In accordance with earlier notation, let τ_k denote the set of tree stumps that include feature k. Then, eq. (B.1) reduces to

$$\begin{aligned}
&w_{\mathbf{X},Y,\hat{y}}(\mathcal{S} \cup \{k\}) - w_{\mathbf{X},Y,\hat{y}}(\mathcal{S}) \\
&= E_{\mathbf{X},Y} \left[2Y \left(\sum_{j \in \tau_k} V_{\mathbf{X},f_j}(\mathcal{S} \cup \{k\}) - V_{\mathbf{X},f_j}(\mathcal{S}) \right) + \left(\sum_{j \in \tau_k} V_{\mathbf{X},f_j}(\mathcal{S}) \right)^2 - \left(\sum_{j \in \tau_k} V_{\mathbf{X},f_j}(\mathcal{S} \cup \{k\}) \right)^2 \right. \\
&\left. + \, 2 \left(\sum_{j \notin \tau_k} V_{\mathbf{X},f_j}(\mathcal{S}) \right) \left(\sum_{j \in \tau_k} V_{\mathbf{X},f_j}(\mathcal{S} \cup \{k\}) - V_{\mathbf{X},f_j}(\mathcal{S}) \right) \right] \\
&= 2Cov \left(Y, \sum_{j \in \tau_k} f_j(X_k) \right) - Var \left(\sum_{j \in \tau_k} f_j(X_k) \right) ,
\end{aligned} \tag{D.1}$$

because all random variables $V_{\mathbf{X},f_j}(\mathcal{S})$ for $j \notin \tau_k$, for every $\mathcal{S}$ are now independent of all $V_{\mathbf{X},f_j}(\mathcal{S})$ and $V_{\mathbf{X},f_j}(\mathcal{S} \cup \{k\})$ for $j \in \tau_k$. Further, for every $j \in \tau_k$, $V_{\mathbf{X},f_j}(\mathcal{S}) =$

$E_{\mathbf{X}}[f_j(\mathbf{X})]$, a constant equal to the expected value of the output of the regression tree f_j, and $E_{\mathbf{X}}[V_{\mathbf{X},f_j}(\mathcal{S} \cup \{k\})] = E_{\mathbf{X}}[f_j(\mathbf{X})]$, since the regression tree f_j only includes feature k. Therefore, the last term in eq. (B.1) vanishes. Observe that, in the case of tree stumps,

$$\begin{aligned}
&E_{\mathbf{X},Y}\left[Y\left(\sum_{j\in\tau_k} V_{\mathbf{X},f_j}(\mathcal{S}\cup\{k\}) - V_{\mathbf{X},f_j}(\mathcal{S})\right)\right] \\
&= E_{\mathbf{X},Y}\left[Y\left(\sum_{j\in\tau_k} f_j(X_k)\right)\right] - E_Y[Y]E_{\mathbf{X}}\left[\sum_{j\in\tau_k} f_j(X_k)\right] = \operatorname{Cov}\left(Y, \sum_{j\in\tau_k} f_j(X_k)\right).
\end{aligned}$$

Likewise,

$$\begin{aligned}
&E_{\mathbf{X},Y}\left[\left(\sum_{j\in\tau_k} V_{\mathbf{X},f_j}(\mathcal{S}\cup\{k\})\right)^2 - \left(\sum_{j\in\tau_k} V_{\mathbf{X},f_j}(\mathcal{S})\right)^2\right] \\
&= E_{\mathbf{X}}\left[\left(\sum_{j\in\tau_k} f_j(X_k)\right)^2\right] - E_{\mathbf{X}}\left[\sum_{j\in\tau_k} f_j(X_k)\right]^2 = \operatorname{Var}\left(\sum_{j\in\tau_k} f_j(X_k)\right).
\end{aligned}$$

Hence, the expression given in eq. (D.1) independent of the subset $\mathcal{S}$. The expression in eq. (D.1) is therefore also equal to the sub-SAGE value, $\hat{\psi}_k$ (or SAGE value). Both the covariance and the variance need to be must be estimated in practice. Given independent test data $(\mathbf{x}_1^0, y_1^0), \ldots, (\mathbf{x}_{N_I}^0, y_{N_I}^0)$, an unbiased estimate is given by

$$\begin{aligned}
\hat{\psi}_k = \frac{1}{N_I^0 - 1}\sum_{i=1}^{N_I^0}\left(y_i^0 - \sum_{i=1}^{N_I} y_i^0\right)\left(\sum_{j\in\tau_k} f_j(x_{i,k}^0) - \sum_{j\in\tau_k} v_{x_{i,k}^0, f_j}(\emptyset)\right) \\
- \frac{1}{N_I^0 - 1}\sum_{i=1}^{N_I^0}\left(\sum_{j\in\tau_k} f_j(x_{i,k}^0) - \sum_{j\in\tau_k} v_{x_{i,k}^0, f_j}(\emptyset)\right)^2.
\end{aligned} \tag{D.2}$$

Appendix E. Sub-SAGE properties related to Shapley values

Symmetry, null player, linearity, monotonicity and efficiency are all properties of Shapley values. Below we investigate whether the same properties apply for sub-SAGE values.

Appendix E.1. Symmetry

Given two features j and k such that $v(\mathcal{S} \cup \{j\}) = v(\mathcal{S} \cup \{k\})$ for all $\mathcal{S} \in \{\mathcal{Q}_j, \mathcal{Q}_k\}$ in which $\{j, k\} \notin \mathcal{S}$. Then their sub-SAGE values indeed are identical, $\psi_j = \psi_k$, and so the symmetry property follows by definition. This means in practice that two perfectly correlated features have equal sub-SAGE values.

Appendix E.2. Dummy property (null player)

Given a feature k where $v(\mathcal{S} \cup \{k\}) = v(\mathcal{S})$ for all $\mathcal{S} \in \mathcal{Q}_k$. Then $\psi_k = 0$, and so the dummy property follows by definition.

Appendix E.3. Linearity

Given two value functions $v(\mathcal{S})$ and $w(\mathcal{S})$, the sub-SAGE value of the sum of the value functions $v(\mathcal{S}) + w(\mathcal{S})$ is equal to the sum of the sub-SAGE for each value function,

$$\psi_k(v + w) = \psi_k(v) + \psi_k(w)\,. \tag{E.1}$$

Appendix E.4. Monotonicity

Consider two models $\hat{f}_1$ and $\hat{f}_2$ used to predict the same relationship $y = f(\mathbf{x})$, for the same features $\mathbf{x}$. If for any feature k we have $v_{\hat{f}_1}(\mathcal{S} \cup \{k\}) - v_{\hat{f}_1}(\mathcal{S}) \geq v_{\hat{f}_2}(\mathcal{S} \cup \{k\}) - v_{\hat{f}_2}(\mathcal{S})$ for all $\mathcal{S} \in \mathcal{Q}_k$, then by definition, $\psi_k^{\hat{f}_1} \geq \psi_k^{\hat{f}_2}$, with $\psi_k^{\hat{f}_1}$ the sub-SAGE value of feature k when applying model $\hat{f}_1$ and $\psi_k^{\hat{f}_2}$ the corresponding sub-SAGE value when applying model $\hat{f}_2$. This means that an adjustment of model $\hat{f}_2$ to $\hat{f}_1$ such that feature k's importance increases also increases its sub-SAGE value. Therefore, the monotonicity property follows by definition.

Appendix E.5. sub-SAGE does not share the efficiency property

Consider the definition of the Shapley value, ϕ_k, applied on a specific value function v:

$$\phi_k = \sum_{\mathcal{S} \subseteq \mathcal{M} \setminus \{k\}} \frac{|\mathcal{S}|!(M - |\mathcal{S}| - 1)!}{M!} \left[v(\mathcal{S} \cup \{k\}) - v(\mathcal{S}) \right] \,. \tag{E.2}$$

The efficiency property for the Shapley value reads

$$\sum_{k=1}^{M} \phi_k = v(\mathcal{M}) - v(\emptyset)\,, \tag{E.3}$$

for M "players". This can be observed more easily by using instead the following formulation of the Shapley value

$$\phi_k = \frac{1}{M!} \sum_R \left[v(s_k(R) \cup \{k\}) - v(s_k(R)) \right] , \tag{E.4}$$

where the sum is over all *orderings* R of the M features, with a total of $M!$ orders. The function $s_k(R)$ maps a given ordering R and a particular feature k to the corresponding subset of features preceding feature k in the specific ordering. For instance, for $\mathcal{M} = \{1, 2, 3\}$, one possible ordering is $R = (2, 3, 1)$ with $s_1(R) = (2, 3)$. We then have

$$\begin{aligned} \sum_{k=1}^{M} \phi_k &= \sum_{k=1}^{M} \frac{1}{M!} \sum_R \left[v(s_k(R) \cup \{k\}) - v(s_k(R)) \right] \\ &= \frac{1}{M!} \sum_R \sum_{k=1}^{M} \left[v(s_k(R) \cup \{k\}) - v(s_k(R)) \right] \\ &= \frac{1}{M!} \sum_R \left(v(\mathcal{M}) - v(\emptyset) \right) \\ &= \frac{1}{M!} M! \left(v(\mathcal{M}) - v(\emptyset) \right) = v(\mathcal{M}) - v(\emptyset), \end{aligned} \tag{E.5}$$

since for a specific ordering R and feature k, in the sum $\sum_{k=1}^{M} \left[v(s_k(R) \cup \{k\}) - v(s_k(R)) \right]$ all terms cancel each other, except $v(\mathcal{M})$ and $v(\emptyset)$.

The sub-SAGE value, ψ_k, for a feature k is not a sum over all subsets $\mathcal{S} \subseteq \mathcal{M} \setminus \{k\}$, but limited to the sets in $\mathcal{Q}_k$,

$$\psi_k(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{\mathcal{S} \in \mathcal{Q}_k} \frac{|\mathcal{S}|!(M - |\mathcal{S}| - 1)!}{3(M - 1)!} \left[v\left(\mathcal{S} \cup \{k\}\right) - v\left(\mathcal{S}\right) \right] , \tag{E.6}$$

and therefore, from the definition in eq. (E.4), is *not* the sum over all orderings R. The sub-SAGE value therefore does not share the efficiency property of the Shapley value.

Appendix F. SHAP computations for fig. 3

Consider this time the SHAP value of a given data generating process, f, with known relationship:

$$\phi_k^{\mathrm{SHAP}}(\mathbf{x}, f) = \sum_{\mathcal{S} \subseteq \mathcal{M} \setminus \{k\}} \frac{|\mathcal{S}|!(M - |\mathcal{S}| - 1)!}{M!} \left[v_{\mathbf{x},f}(\mathcal{S} \cup \{k\}) - v_{\mathbf{x},f}(\mathcal{S}) \right] , \tag{F.1}$$

By applying the data generating process, f, explained in Section 5, the exact SHAP value of feature 1 can be computed by partitioning in the subsets $\mathcal{S}$ *not including* feature 2, as well as those *including* feature 2. For all $\mathcal{S}$ not including feature 2, and by using the result in Appendix B in Aas et al. (2021):

$$v_{\mathbf{x}_i,f}(\mathcal{S} \cup \{k\}) - v_{\mathbf{x}_i,f}(\mathcal{S}) = a_1(x_{i,1} - E[X_1]) + a_{21}E[e^{X_2}](x_{i,1} - E[X_1]),$$

independent of the subset $\mathcal{S}$ used. Of all $\mathcal{S} \subseteq \mathcal{M} \setminus \{1\}$, a half of them will not include feature 2, and the sum of the corresponding Shapley weights are given by:

$$\begin{aligned}
&\sum_{|\mathcal{S}|=0}^{M-2} \frac{|\mathcal{S}|!(M-|\mathcal{S}|-1}{M!} \binom{M-2}{|\mathcal{S}|} = \sum_{|\mathcal{S}|=0}^{M-2} \frac{|\mathcal{S}|!(M-|\mathcal{S}|-1}{M!} \frac{(M-2)!}{|\mathcal{S}|!(M-2-|\mathcal{S}|!)} \\
&= \sum_{|\mathcal{S}|=0}^{M-2} \frac{1}{M} - \frac{1}{M(M-1)} \sum_{|\mathcal{S}|=0}^{M-2} |\mathcal{S}| = \frac{1}{2}.
\end{aligned}$$

For all $\mathcal{S}$ including feature 2:

$$v_{\mathbf{x}_i,f}(\mathcal{S} \cup \{k\}) - v_{\mathbf{x}_i,f}(\mathcal{S}) = a_1(x_{i,1} - E[X_1]) + a_{21}e^{x_{i,2}}(x_{i,1} - E[X_1]).$$

As the sum of the Shapley weights are equal to one, the sum of the Shapley weights for these $\mathcal{S}$ must also be 1/2. Hence, the SHAP value of feature 1 is given by:

$$\begin{aligned}
\phi_{i,1}(\mathbf{x}_i) &= \frac{1}{2}(a_1(x_{i,1} - E[X_1]) + a_{21}E[e^{X_2}](x_{i,1} - E[X_1])) \\
&+ \frac{1}{2}(a_1(x_{i,1} - E[X_1]) + a_{21}e^{x_{i,2}}(x_{i,1} - E[X_1])) \\
&= a_1(x_{i,1} - E[X_1]) + a_{21}E[e^{X_2}](x_{i,1} - E[X_1]) \\
&+ \frac{1}{2}a_{21}x_{i,1}(e^{x_{i,2}} - E[e^{X_2}]) - \frac{1}{2}a_{21}E[X_1](e^{x_{i,2}} - E[e^{X_2}]).
\end{aligned} \tag{F.2}$$

In the exact same manner one can show that:

$$\begin{aligned}
\phi_{i,2}(\mathbf{x}_i) &= \frac{1}{2}(a_2(x_{i,2} - E[X_2]) + a_{21}E[X_1](e^{x_{i,2}} - E[e^{X_2}]) \\
&+ \frac{1}{2}(a_2(x_{i,2} - E[X_2]) + a_{21}x_{i,1}(e^{x_{i,2}} - E[e^{X_2}])) \\
&= a_2(x_{i,2} - E[X_2]) + a_{21}E[X_1](e^{x_{i,2}} - E[e^{X_2}]) \\
&+ \frac{1}{2}a_{21}x_{i,1}(e^{x_{i,2}} - E[e^{X_2}]) - \frac{1}{2}a_{21}E[X_1](e^{x_{i,2}} - E[e^{X_2}]).
\end{aligned} \tag{F.3}$$

$$
\begin{aligned}
&\phi_{i,6} = \frac{1}{2}(a_6 E[X_5](I(x_{i,6} > 7) - E[I(X_6 > 7)]) \\
&\frac{1}{2}(a_6 x_{i,5}(I(x_{i,6} > 7) - E[I(X_6 > 7)])) \\
&= a_6 E[X_5](I(x_{i,6} > 7) - E[I(X_6 > 7)]) \\
&+ \frac{1}{2}a_6 I(x_{i,6} > 7)(x_{i,5} - E[X_5]) - \frac{1}{2}a_6 E[I(X_6 > 7)](x_{i,5} - E[X_5]).
\end{aligned} \tag{F.4}
$$

References

Aas, K., Jullum, M., Løland, A., 2021. Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. Artificial Intelligence 298.

Bycroft, C., Freeman, C., Petkova, D., Band, G., Elliott, L.T., Sharp, K., Motyer, A., Vukcevic, D., Delaneau, O., O'Connell, J., et al., 2018. The uk biobank resource with deep phenotyping and genomic data. Nature 562, 203–209.

Chen, T., Guestrin, C., 2016. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16 , 785–794.

Covert, I., Lundberg, S., Lee, S.I., 2020a. Explaining by removing: A unified framework for model explanation. `arXiv:2011.14878`.

Covert, I., Lundberg, S., Lee, S.I., 2020b. Understanding global feature contributions with additive importance measures. `arXiv:2004.00668`.

Efron, B., 1987. Better Bootstrap Confidence Intervals. Journal of the American Statistical Association 82, 171–185.

Efron, B., Tibshirani, R.J., 1994. An Introduction to the Bootstrap. Chapman & Hall/CRC.

Fryer, D., Strümke, I., Nguyen, H., 2021a. Shapley values for feature selection: The good, the bad, and the axioms. `arXiv:2102.10936`.

Fryer, D.V., Strumke, I., Nguyen, H., 2021b. Model independent feature attributions: Shapley values that uncover non-linear dependencies. PeerJ Computer Science 7, e582.

Gagliano Taliun, S.A., VandeHaar, P., Boughton, A.P., Welch, R.P., Taliun, D., Schmidt, E.M., Zhou, W., Nielsen, J.B., Willer, C.J., Lee, S., Fritsche, L.G., Boehnke, M., Abecasis, G.R., 2020. Exploring and visualizing large-scale genetic associations by using PheWeb. Nature Genetics 52. URL: `https://pheweb.org/UKB-TOPMed/`.

Goeman, J.J., Solari, A., 2014. Multiple hypothesis testing in genomics. Statistics in Medicine 33, 1946–1978.

Huettner, F., Sunder, M., 2012. Axiomatic arguments for decomposiing goodness of fit according to Shapley and Owen values. Electronic Journal of Statistics 6, 1239–1250.

Johnsen, P.V., Riemer-Sørensen, S., DeWan, A.T., Cahill, M.E., Langaas, M., 2021. A new method for exploring gene–gene and gene–environment interactions in GWAS with tree ensemble methods and SHAP values. BMC Bioinformatics 22.

Karlsson, T., Rask-Andersen, M., Pan, G., Höglund, J., Wadelius, C., Ek, W.E., Johansson, Å., 2019. Contribution of genetics to visceral adiposity and its relation to cardiovascular and metabolic disease. Nature medicine 25, 1390–1395.

Karoui, N.E., Purdom, E., 2018. Can We Trust the Bootstrap in High-dimensions? The Case of Linear Models. Journal of Machine Learning Research 19, 66.

Keinan, A., Hilgetag, C.C., Meilijson, I., Ruppin, E., 2003. Fair attribution of functional contribution in artificial and biological networks. Neural Computation 16, 1887–1915.

Kwon, Y., Rivas, M.A., Zou, J., 2021. Efficient computation and analysis of distributional Shapley values. `arXiv:2007.01357`.

Locke, A.E., Kahali, B., Berndt, S.I., et al., 2015. Genetic studies of body mass index yield new insights for obesity biology. Nature 518, 197–206.

Lundberg, S.M., Erion, G., Chen, H., DeGrave, A., Prutkin, J.M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., Lee, S.I., 2020. From local explanations to global understanding with explainable AI for trees. Nature Machine Intelligence 2.

Lundberg, S.M., Erion, G.G., Lee, S.I., 2019. Consistent individualized feature attribution for tree ensembles. `arXiv:1802.03888`.

Lundberg, S.M., Lee, S.I., 2017. A Unified Approach to Interpreting Model Predictions, in: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 30. Curran Associates, Inc., pp. 4765–4774.

Moehle, N., Boyd, S., Ang, A., 2021. Portfolio performance attribution via Shapley value. arXiv:2102.05799.

Redelmeier, A., Jullum, M., Aas, K., 2020. Explaining predictive models with mixed features using Shapley values and conditional inference trees. arXiv:2007.01027.

Sellereite, N., Jullum, M., 2019. shapr: An R-package for explaining machine learning models with dependence-aware Shapley values. Journal of Open Source Software 5, 2027. URL: https://doi.org/10.21105/joss.02027, doi:10.21105/joss.02027.

Shapley, L.S., 1953. A value for n-person games, in: Contributions to the Theory of Games (AM-28), Volume II.

Song, E., Nelson, B., Staum, J., 2016. Shapley effects for global sensitivity analysis: Theory and computation. SIAM/ASA Journal on Uncertainty Quantification 4, 1060–1083. doi:10.1137/15M1048070.

Strumbelj, E., Kononenko, I., 2010. An efficient explanation of individual classifications using game theory. Journal of Machine Learning Research 11, 1–18. doi:10.1145/1756006.1756007.

Strumbelj, E., Kononenko, I., 2013. Explaining prediction models and individual predictions with feature contributions. Knowledge and Information Systems 41, 647–665. doi:10.1007/s10115-013-0679-x.

Sudlow, C., Gallacher, J., Allen, N., Beral, V., Burton, P., Danesh, J., Downey, P., Elliott, P., Green, J., Landray, M., et al., 2015. Uk biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. PLoS medicine 12, e1001779.

Visscher, P.M., Wray, N.R., Zhang, Q., Sklar, P., McCarthy, M.I., Brown, M.A., Yang, J., 2017. 10 Years of GWAS Discovery: Biology, Function, and Translation. American Journal of Human Genetics 101, 5–22.

Young, H.P., 1985. Monotonic solutions of cooperative games. International Journal of Game Theory 14, 65–72.

Zhou, W., Nielsen, J.B., Fritsche, L.G., Dey, R., Gabrielsen, M.E., Wolford, B.N., LeFaive, J., VandeHaar, P., Gagliano, S.A., Gifford, A., Bastarache, L.A., Wei, W.Q., Denny, J.C., Lin, M., Hveem, K., Kang, H.M., Abecasis, G.R., Willer, C.J., Lee, S., 2018. Efficiently controlling for case-control imbalance and sample relatedness in large-scale genetic association studies. Nature Genetics 50.

ISBN 978-82-326-5639-4 (printed ver.)
ISBN 978-82-326-6718-5 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

NTNU
Norwegian University of
Science and Technology