

Untitled

PHS 7095, Spring 2024 - Final Report

Kline Dubose, Haojia Li

05/05/2024

1 Introduction

- Non-standard because the parameter under the null hypothesis is on the boundary of the parameter space and the response variables are not independent under the alternative.
- Special case of variance component of familial relatedness in linear mixed models.

2 Likelihood Ratio Tests in Linear Mixed Models with One Variance Component

Crainiceanu and Ruppert (2004) derived both the finite sample distribution and the asymptotic distribution of the LRT and RLRT statistics for testing the null hypothesis that the variance component is 0 in a linear mixed model (LMM) with one variance component. They used weak assumptions on eigenvalues of certain design matrices, and released the hypothesis of i.i.d. data in the restrictive assumptions by Self and Liang (1987), that the response variable vector can be partitioned into independent and identically distributed subvectors and the number of independent subvectors tends to ∞ .

2.1 Model and hypotheses setup

Consider an LMM with one variance component

$$\mathbf{Y} = X\beta + Z\mathbf{b} + \epsilon, \quad E \begin{pmatrix} \mathbf{b} \\ \epsilon \end{pmatrix} = \begin{pmatrix} \mathbf{0}_K \\ \mathbf{0}_n \end{pmatrix}, \quad \text{cov} \begin{pmatrix} \mathbf{b} \\ \epsilon \end{pmatrix} = \begin{pmatrix} \sigma_b^2 \Sigma & \mathbf{0} \\ \mathbf{0} & \sigma_\epsilon^2 I_n \end{pmatrix}, \quad (1)$$

where,

- \mathbf{Y} is the $n \times 1$ vector of observations,
- X is the $n \times p$ design matrix for the fixed effects,
- Z is the $n \times K$ design matrix for the random effects,
- β is a p -dimensional vector of fixed effects parameters,
- \mathbf{b} is a K -dimensional vector of random effects,
- (\mathbf{b}, ϵ) has a normal distribution.

Under these conditions it follows that

$$E(\mathbf{Y}) = X\beta \text{ and } \text{Cov}(\mathbf{Y}) = \sigma_\epsilon^2 V_\lambda,$$

where

- $\lambda = \sigma_b^2 / \sigma_\epsilon^2$, which can be considered a signal-to-noise ratio,
- $V_\lambda = I_n + \lambda Z \Sigma Z'$.

In this context, $\sigma_b^2 = 0$ if and only if $\lambda = 0$, and the parameter space for λ is $[0, \infty)$.

The authors proposed a general form of the null hypothesis (H_0) and alternative hypothesis (H_A):

$$\begin{aligned} H_0 : \beta_{p+1-q} &= \beta_{p+1-q}^0, \dots, \beta_p = \beta_p^0, & \sigma_b^2 &= 0 \text{ (or equivalently } \lambda = 0) \\ H_A : \beta_{p+1-q} &\neq \beta_{p+1-q}^0, \dots, \beta_p \neq \beta_p^0, & \sigma_b^2 &> 0 \text{ (or equivalently } \lambda > 0) \end{aligned}$$

where the q denotes the number of fixed effects parameters constrained under H_0 . An application of this method with $q > 0$ is using LRTs and RLRTs for testing a polynomial fit against a general alternative described by penalized splines (P-splines), in Section 5 of the paper.

However, we will only focus on the particular case of $q = 0$:

$$H_0 : \sigma_b^2 = 0 \text{ } (\lambda = 0) \text{ vs. } H_A : \sigma_b^2 > 0 \text{ } (\lambda > 0)$$

2.2 Finite sample distribution of LRT and RLRT

Twice the log-likelihood ratio statistic for model (1) is

$$2 \log L(\beta, \lambda, \sigma_\epsilon^2) = -\log(\sigma_\epsilon^2) - \log |V_\lambda| - \frac{(Y - X\beta)' V_\lambda^{-1} (Y - X\beta)}{\sigma_\epsilon^2} \quad (2)$$

and the LRT is defined as

$$LRT_n = 2 \sup_{H_A} \{L(\beta, \lambda, \sigma_\epsilon^2)\} - 2 \sup_{H_0} \{L(\beta, \lambda, \sigma_\epsilon^2)\}.$$

Under the alternative hypothesis, by fixing λ and solving the first-order maximum conditions for β and σ_ϵ^2 , we obtain the profile likelihood estimates

$$\hat{\beta}(\lambda) = (X' V_\lambda^{-1} X)^{-1} X' V_\lambda^{-1} \mathbf{Y}, \quad \hat{\sigma}_\epsilon^2(\lambda) = \frac{(\mathbf{Y} - X \hat{\beta}(\lambda))' V_\lambda^{-1} (\mathbf{Y} - X \hat{\beta}(\lambda))}{n}.$$

Plugging these estimates back into the log-likelihood function, we obtain the profile log-likelihood function

$$L^{K,n}(\lambda) = -\log |V_\lambda| - n \log(\mathbf{Y}' P_\lambda' V_\lambda^{-1} P_\lambda \mathbf{Y}),$$

where $P_\lambda = I_n - X(X' V_\lambda^{-1} X)^{-1} X' V_\lambda^{-1}$ and $P_0 = I_n - X(X' X)^{-1} X'$ under the null hypothesis when the model becomes a standard linear regression and $V_0 = I_n$.

The LRT statistic can be written as

$$LRT_n = \sup_{\lambda \geq 0} \{n \log(\mathbf{Y}' P_0 \mathbf{Y}) - n \log(\mathbf{Y}' P'_\lambda V_\lambda^{-1} P_\lambda \mathbf{Y}) - \log |V_\lambda|\}, \quad (3)$$

which has the spectral decomposition given by Theorem 1 shown below.

Theorem 1. If $\mu_{s,n}$ and $\xi_{s,n}$ are the K eigenvalues of the $K \times K$ matrices $\Sigma^{1/2} Z' P_0 Z \Sigma^{1/2}$ and $\Sigma^{1/2} Z' Z \Sigma^{1/2}$ respectively, where $P_0 = I_n - X(X'X)^{-1}X'$, then

$$LRT_n \stackrel{\mathcal{D}}{=} \sup_{\lambda \geq 0} f_n(\lambda) = \sup_{\lambda \geq 0} \left\{ n \log \frac{N_n(\lambda) + D_n(\lambda)}{D_n(\lambda)} - \sum_{s=1}^K \log(1 + \lambda \xi_{s,n}) \right\}, \quad (4)$$

where the notation $\stackrel{\mathcal{D}}{=}$ denotes equality in distribution and

$$N_n(\lambda) = \sum_{s=1}^K \frac{\lambda \mu_{s,n}}{1 + \lambda \mu_{s,n}}, \quad D_n(\lambda) = \sum_{s=1}^K \frac{w_s^2}{1 + \lambda \mu_{s,n}} + \sum_{s=K+1}^{n-p} w_s^2.$$

w_s for $s = 1, \dots, n-p$ are independent $N(0, 1)$.

The distribution described in equation (4) only depends on the eigenvalues $\mu_{s,n}$ in the first part and $\xi_{s,n}$ in the second part of the two $K \times K$ matrices. Taking the advantage of the spectral decomposition, the eigenvalues would only need to be calculated once before the simulation starts the iterative process. The speed depends on the number of random effects K instead of the number of observations n . The steps of the algorithm to simulate the null finite sample distribution of LRT_n are presented in Section 4.

The probability of having a local maximum of the profile likelihood at $\lambda = 0$ is

$$\text{pr} \left(\frac{\sum_{s=1}^K \mu_{s,n} w_s^2}{\sum_{s=1}^{n-p} w_s^2} \leq \frac{1}{n} \sum_{s=1}^K \xi_{s,n} \right)$$

where w_s are independent $N(0, 1)$ random variables.

This is the exact probability of a local maximum at 0 and provides an excellent approximation for the probability of a global maximum at $\lambda = 0$.

Similarly, twice the restricted profile log-likelihood function is

$$2l^{K,n}(\lambda) = -\log |V_\lambda| - \log |X' V_\lambda^{-1} X| - (n-p) \log(\mathbf{Y}' P'_\lambda V_\lambda^{-1} P_\lambda \mathbf{Y}).$$

Then, under the null hypothesis $H_0 : \sigma_b^2 = 0$ ($\lambda = 0$),

$$RLRT_n = \sup_{\lambda \geq 0} \left\{ (n-p) \log \frac{N_n(\lambda) + D_n(\lambda)}{D_n(\lambda)} - \sum_{s=1}^K \log(1 + \lambda \mu_{s,n}) \right\},$$

and the probability of having a local maximum at $\lambda = 0$ is

$$\text{pr} \left(\frac{\sum_{s=1}^K \mu_{s,n} w_s^2}{\sum_{s=1}^{n-p} w_s^2} \leq \frac{1}{n-p} \sum_{s=1}^K \mu_{s,n} \right).$$

Proof of Theorem 1

Starting with equation (3), one can easily show that $\log|V_\lambda| = -\sum_{s=1}^K \log(1 + \lambda\xi_{s,n})$.

Also, from Kuo, 1999 and Patterson and Thompson, 1971 there exists an $n \times (n-p)$ matrix W such that $W'W = I_{n-p}$, $WW' = P_0$, $W'V_\lambda W = \text{diag}\{(1 + \lambda\mu_{s,n})\}$, and

$$\mathbf{Y}'P'_\lambda V_\lambda^{-1}P_\lambda \mathbf{Y} = \mathbf{Y}'W \text{diag}\{(1 + \lambda\mu_{s,n})^{-1}\}W'\mathbf{Y}.$$

Denote by $w = W'\mathbf{Y}/\sigma_\epsilon$ and note that under the null hypothesis

$$E[w] = W'X\beta/\sigma_\epsilon, \quad \text{Cov}[w] = I_{n-p}.$$

We now show that $E[w] = 0$. Denote by $A = W'X$ and observe that $WA = P_0X = 0$, and that $W'WA = 0$. This shows that $A = 0$, that $W'X = 0$. It now follows that $w = (w_1, \dots, w_{n-p})$ is an $n-p$ dimensional random vector with i.i.d. $N(0, 1)$ components. Putting all these together it follows that:

$$L^{K,n}(\lambda) = -n \log \left\{ \sigma_\epsilon^2 \left(\sum_{s=1}^K \frac{w_s^2}{1 + \lambda\mu_{s,n}} + \sum_{s=K+1}^{n-p} w_s^2 \right) \right\} - \sum_{s=1}^K \log(1 + \lambda\xi_{s,n}),$$

where we used the fact that at most K eigenvalues $\mu_{s,n}$ and $\xi_{s,n}$ are not zero. In particular

$$L^{K,n}(0) = -n \log \left\{ \sigma_\epsilon^2 \left(\sum_{s=1}^{n-p} w_s^2 \right) \right\},$$

which is a standard result in regression analysis. Therefore we can write

$$L^{K,n}(\lambda) - L_{k,n}(0) = n \log\{1 + U_n(\lambda)\},$$

where $U_n(\lambda) = N_n(\lambda)/D_n(\lambda)$ and

$$N_n(\lambda) = \sum_{s=1}^K \frac{\lambda\mu_{s,n}}{1 + \lambda\mu_{s,n}} w_s^2, \quad D_n(\lambda) = \sum_{s=1}^K \frac{w_s^2}{1 + \lambda\mu_{s,n}} + \sum_{s=K+1}^{n-p} w_s^2.$$

2.3 Asymptotic distribution of LRT and RLRT

Asymptotic theory for a null hypothesis on the boundary of the parameter space developed for IID data suggests the asymptotic result

$$LRT_n \Rightarrow U_+^2,$$

where U is an $N(0, 1)$ random variable and $U_+^2 = U^2 I(U > 0)$ with $I(\cdot)$ the indicator function. The distribution of U_+^2 is a $0.5\chi_0^2 : 0.5\chi_1^2$ mixture between a χ_0^2 - (Dirac distribution at 0) and a χ_1^2 -distribution.

In contrast, Crainiceanu and Ruppert described a relationship between the asymptotic distributions of test statistics and the asymptotic behavior of the eigenvalues $\mu_{s,n}$ and $\xi_{s,n}$, which is provided in Theorem 2. This asymptotic distribution is generally different from the $0.5\chi_0^2 : 0.5\chi_1^2$ mixture, and differences can be severe.

Theorem 2. Assume that hypothesis H_0 in expression (2) is true. Suppose that there is an $\alpha \geq 0$ so that for every s the K eigenvalues $\mu_{s,n}$ and $\xi_{s,n}$ of matrices $\Sigma^{1/2}Z'P_0Z\Sigma^{1/2}$ and $\Sigma^{1/2}Z'Z\Sigma^{1/2}$ respectively satisfy $\lim_{n \rightarrow \infty} (n^{-\alpha}\mu_{s,n}) = \mu_s$ and $\lim_{n \rightarrow \infty} n^{-\alpha}\xi_{s,n} = \xi_s$, where not all ξ_s are 0. Then

$$LRT_n \Rightarrow \sup_{d \geq 0} \{LRT_\infty(d)\} = \sup_{d \geq 0} \left\{ \sum_{s=1}^K \frac{d\mu_s}{1+d\mu_s} w_s^2 - \sum_{s=1}^K \log(1+d\xi_s) \right\},$$

where \Rightarrow denotes weak convergence.

Under the same assumptions as in theorem 2, if $H_0 : \sigma_b^2 = 0$ ($\lambda = 0$) is true then

$$RLRT_n \Rightarrow \sup_{d \geq 0} \{RLRT_\infty(d)\} = \sup_{d \geq 0} \left\{ \sum_{s=1}^K \frac{d\mu_s}{1+d\mu_s} w_s^2 - \sum_{s=1}^K \log(1+d\mu_s) \right\}.$$

When $q = 0$, LRT and RLRT statistics have probability mass at 0. Crainiceanu, Ruppert and Vogelsang (2003) showed that this mass can be very large for LRT_∞ and $RLRT_\infty$ and is equal to the null probability that $LRT_\infty(\cdot)$ and $RLRT_\infty(\cdot)$ have a global maximum at 0. The latter is well approximated by the null probability of having a local maximum at $\lambda = 0$.

The probability of having a local maximum at $d = 0$ for $LRT_\infty(\cdot)$ or $RLRT_\infty(\cdot)$ is

$$\text{pr} \left(\sum_{s=1}^K \mu_s w_s^2 \leq \sum_{s=1}^K \xi_s \right) \text{ or } \text{pr} \left(\sum_{s=1}^K \mu_s w_s^2 \leq \sum_{s=1}^K \mu_s \right),$$

where w_s are IID $N(0, 1)$ random variables.

Proof of Theorem 2

We continue to use notations from the proof of theorem 1. $LRT_n = \sup_{\lambda \geq 0} LRT_n(\lambda)$ where

$$LRT_n(\lambda) = n \log\{1 + U_n(\lambda)\} - \sum_{s=1}^K \log(1 + \lambda \xi_{s,n}),$$

Denote now by $f_n(d) = n \log\{1 + U_n(n^{-\alpha}d)\} - \sum_{s=1}^K \log(1 + dn^{-\alpha}\xi_{s,n})$ and we will show that

$$\sup_{d \geq 0} f_n(d) \Rightarrow \sup_{d \geq 0} LRT_\infty(d).$$

This proof consists of two steps:

1. Prove that $f_n(\cdot)$ converges weakly to $LRT_\infty(\cdot)$ on the space $\mathcal{C}[0, \infty)$ of continuous functions with support $[0, \infty)$.
2. Prove that a Continuous Mapping Theorem type result holds for the $\sup_{d \geq 0} f_n(d)$.

We show the weak convergence for any $\mathcal{C}[0, M]$. Denote $f(d) = LRT_\infty(d)$, $\eta_{s,n} = n^{-\alpha}\mu_{s,n}$, $\zeta_{s,n} = n^{-\alpha}\xi_{s,n}$, etc. Note that $\lim_{n \rightarrow \infty} \eta_{s,n} = \mu_s$, and $\lim_{n \rightarrow \infty} \zeta_{s,n} = \xi_s$. We first establish the finite dimensional convergence of $f_n(d)$ to $f(d)$ and then we prove that $f_n(d)$ is a tight sequence in $\mathcal{C}[0, M]$.

To show finite dimensional convergence it is sufficient to show that for a fixed d the convergence is almost sure. Note that

$$n \log\{1 + U_n(n^{-\alpha}d)\} = nU_n(n^{-\alpha}d) + nR(n^{-\alpha}d),$$

where $R(\cdot)$, $U_n(\cdot)$, $N_n(\cdot)$ and $D_n(\cdot)$ were defined earlier. It follows immediately that almost surely

$$\lim_{n \rightarrow \infty} nU_n(n^{-\alpha}d) = \sum_{s=1}^K \frac{d\mu_s}{1 + d\mu_s} w_s^2.$$

Because $nR\{U_n(n^{-\alpha}d)\} = \{nU_n(n^{-\alpha}d)\}\{R(U_n(n^{-\alpha}d)/U_n(n^{-\alpha}d))\}$, it follows that $nR(U_n(n^{-\alpha}d))$ converges to zero almost surely ($\lim_{x \rightarrow 0} R(x)/x = 0$). Note that $\lim_{n \rightarrow \infty} \sum_{s=1}^K \log(1 + d\zeta_{s,n}) = \sum_{s=1}^K \log(1 + d\zeta_s)$ for every fixed d .

To show that $f_n(d)$ form a tight sequence it is sufficient to show that for every ϵ and η and strictly positive, there exist $\delta = \delta(\epsilon, \eta)$, $0 < \delta < 1$ and $n_0 = n_0(\epsilon, \delta)$ such that for $n \geq n_0$

$$\frac{1}{\delta} P \left(\sup_{t \leq u \leq t+\delta} |f_n(u) - f_n(t)| \geq \epsilon \right) < \eta.$$

Observe first that

$$|f_n(u) - f_n(t)| \leq n \log \left\{ \frac{D_n(n^{-\alpha}t)}{D_n(n^{-\alpha}u)} \right\} + \sum_{s=1}^K \log \frac{1 + u\zeta_{s,n}}{1 + t\zeta_{s,n}}.$$

And because $\log(1 + x) < x$ for every $x > 0$ we obtain the following inequalities

$$\log \left\{ \frac{D_n(n^{-\alpha}t)}{D_n(n^{-\alpha}u)} \right\} \leq \frac{D_n(n^{-\alpha}t) - D_n(n^{-\alpha}u)}{D_n(n^{-\alpha}u)} \leq (s - t)C \frac{\sum_{s=1}^K w_s^2}{\sum_{s=K+1}^{n-p} w_s^2},$$

where $C > 0$ is a constant so that $n\zeta_{s,n}/(n - p - K) \leq C$ for every s and n . It follows that the following inequality holds

$$n \log \left\{ \frac{D_n(n^{-\alpha}t)}{D_n(n^{-\alpha}u)} \right\} \leq (u - t)CKF_{K,n},$$

where $F_{K,n}$ is a random variable with an F distribution with $(K, n - p - K)$ degrees of freedom. Similarly,

$$\sum_{s=1}^K \log \left(\frac{1 + u\zeta_{s,n}}{1 + t\zeta_{s,n}} \right) \leq (u - t)CK.$$

We conclude that

$$P \left\{ \sup_{t \leq u \leq t+\delta} |f_n(u) - f_n(t)| \geq \epsilon \right\} \leq P \{ F_{K,n} \geq \epsilon(CK\delta)^{-1} - 1 \},$$

and it is sufficient to find δ, n_0 such that for every $n \geq n_0$ the c.d.f. $H_{K,n}$ of $F_{K,n}$ satisfies

$$1 - H_{K,n} \left(\frac{\epsilon}{CK\delta} - 1 \right) \leq \eta\delta. \quad (5)$$

If H_K is the c.d.f. of a χ^2 distribution with K degrees of freedom, then for every x , $\lim_{n \rightarrow \infty} H_{K,n}(x) = H_K(K_x)$. Because (using for example l'Hospital rule and the pdf of a χ^2 distribution with K degrees of freedom)

$$\lim_{\delta \downarrow 0} \left\{ 1 - H_K \left(\frac{\epsilon}{C\delta} - K \right) \right\} / \left\{ \frac{\eta\delta}{2} \right\} = 0,$$

one can find $\delta = \delta(\epsilon, \eta)$, $\delta < 1$ with $\epsilon/C\delta - K > 0$ such that $1 - H_K(\epsilon/C\delta - K) \leq \eta\delta/2$. Because of the convergence of $H_{K,n}$ to H_K , one can find $n_0 = n_0(\epsilon, \eta)$ so that for $n \geq n_0$ the following inequality holds

$$\left| H_{K,n} \left(\frac{\epsilon}{C\delta} - K \right) - H_K \left(\frac{\epsilon}{C\delta} - K \right) \right| \leq \frac{\eta\delta}{2},$$

which finishes the proof of the inequality in equation (5). We conclude that $f_n(d)$ converges weakly to $f(d)$ on $\mathcal{C}[0, M]$ for each M , and therefore on $\mathcal{C}[0, \infty)$.

We want to show now that $\sup_{d \geq 0} f_n(d) \Rightarrow \sup_{d \geq 0} f(d)$. First we find a random variable $T_{K,n}$ such that

$$\sup_{d \geq 0} f_n(d) = \max_{d \in [0, T_{K,n}]} f_n(d).$$

Note first that $f_n(d) = W_q$ for every n . Also, using again the inequality $\log(1+x) \leq x$ for $x \geq 0$ it is easy to prove that

$$f_n(d) \leq n \frac{\sum_{s=1}^K w_s^2}{\sum_{s=K+1}^{n-p} w_s^2} - K \log(1+dm) = \frac{nK}{n-p-K} F_{K,n} - K \log(1+dm)$$

Denote by

$$T_{K,n} = \frac{1}{m} \left\{ \exp \left(\frac{n}{n-p-K} F_{K,n} \right) - 1 \right\},$$

and observe that for $d > T_{K,n}$ we have $f_n(d) < W_q$ which shows that $T_{K,n}$ has the desired property. Observe now that for every $M > 0$ and for every $t \geq 0$

$$\text{pr} \left\{ \sup_{d \geq 0} f_n(d) \leq t \right\} \leq \text{pr} \left\{ \max_{d \in [0, M]} f_n(d) \leq t \right\}.$$

Taking $\lim \sup$ for $n \rightarrow \infty$ one obtains

$$\limsup_{n \rightarrow \infty} \text{pr} \left\{ \sup_{d \geq 0} f_n(d) \leq t \right\} \leq \limsup_{n \rightarrow \infty} \text{pr} \left\{ \max_{d \in [0, M]} f_n(d) \leq t \right\}.$$

Because $f_n(d) \Rightarrow f(d)$ on $\mathcal{C}[0, M]$ and by extension on $\mathcal{C}[0, \infty)$ one can apply the Continuous Mapping Theorem for the right hand side of the inequality and we obtain

$$\lim_{n \rightarrow \infty} \text{pr} \left\{ \max_{d \in [0, M]} f_n(d) \leq t \right\} = \text{pr} \left\{ \max_{d \in [0, M]} f(d) \leq t \right\}. \quad (6)$$

Using that for any two events A and B , $P(A \cap B) \geq P(A) - P(B^C)$, we obtain the following sequence of relations:

$$\begin{aligned} \Pr \left\{ \sup_{d \geq 0} f_n(d) \leq t \right\} &\geq \Pr \left\{ \sup_{d > 0} f_n(d) \leq t, T_{K,n} < M \right\} \\ &= \Pr \left\{ \max_{d \in [0, M]} f_n(d) \leq t, T_{K,n} < M \right\} \\ &\geq \Pr \left\{ \max_{d \in [0, M]} f_n(d) \leq t \right\} - \Pr \{ T_{K,n} \geq M \}, \end{aligned}$$

Taking the limit when $n \rightarrow \infty$ in the first and last expressions we obtain:

$$\liminf_{n \rightarrow \infty} \Pr \left(\sup_{d \geq 0} f_n(d) \leq t \right) \geq \Pr \left\{ \sup_{d \in [0, M]} f(d) \leq t \right\} - \Pr(T_K \geq M),$$

where we used equation (6) and $T_K = \left\{ \exp \left(\sum_{s=1}^K w_s^2 / K \right) - 1 \right\} / m$. Consider now a sequence of integers $M \rightarrow \infty$. Then $\lim_{M \rightarrow \infty} \Pr(T_K \geq M) = 0$. Therefore if we prove that:

$$\lim_{M \rightarrow \infty} \Pr \left\{ \max_{d \in [0, M]} f_n(d) \leq t \right\} = \Pr \left\{ \sup_{d \geq 0} f(d) \leq t \right\},$$

then it follows that $\lim_{n \rightarrow \infty} \Pr(\sup_{d > 0} f_n(d) \leq t)$ exists and:

$$\lim_{n \rightarrow \infty} \Pr \left\{ \sup_{d \geq 0} f_n(d) \leq t \right\} = \Pr \left\{ \sup_{d > 0} f(d) \leq t \right\},$$

proving that:

$$\sup_{d \geq 0} f_n(d) \Rightarrow \sup_{d \geq 0} f(d).$$

Now that

$$LRT_n \Rightarrow \sup_{d \geq 0} f(d) = \sup_{d \geq 0} LRT_\infty(d).$$

3 Proofs of Lemma 3, Theorem 1, and Corollary 1

Before we begin, in this scenario, our null hypothesis is $\lambda = 0$ where λ is the effect size of familial relatedness (FR) and the alternative is $\lambda > 0$.

3.1 Lemma 3

Suppose that $\sup_{s \in \Omega_0} \{\rho_s\} \leq B$ are bounded. Under the null hypothesis, (i) both $F_n(\lambda)$ and $G_n(\lambda)$ uniformly converge in probability to $F_0(\lambda) = \sum_{j=0}^m q_j \log(w_j) - \log(\sum_{j=0}^m q_j w_j)$ over $\lambda \in [-\delta, T]$ for any $0 < \delta < \frac{1}{B \vee \max_j \{\phi_j\}}$ and $0 < T < +\infty$; (ii) $F_0(\lambda)$ achieves its unique maximum at $\lambda = 0$; (iii) $\hat{\lambda}_n \rightarrow^p 0$ and $\hat{\lambda}_n^r \rightarrow^p 0$.

i) Essentially we will be showing that:

$$\begin{aligned} F_n(\lambda) &\rightarrow^p F_0(\lambda) \\ G_n(\lambda) &\rightarrow^p F_0(\lambda) \end{aligned}$$

where:

$$F_0(\lambda) = \sum_{j=0}^m \log(w_j) - \log\left(\sum_{j=0}^m q_j w_j\right)$$

Additionally, note that ϕ_j is the set of all distinct non-zero eigenvalues of the FR correlation matrix.

Let's start with rewriting $F_n(\lambda)$:

$$\begin{aligned} F_n(\lambda) &= \log\left(\frac{\sum_{i=1}^{n-p} u_i^2}{\sum_{j=0}^m f'_j w_j U_j + \sum_{s \in \Omega_0} v_s u_s^2}\right) + \frac{1}{n} \sum_{j=1}^m f_j \log(w_j) \\ &= \log\left(\frac{n-p}{n-p} \cdot \frac{\sum_{i=1}^{n-p} u_i^2}{\sum_{j=0}^m f'_j w_j U_j + \sum_{s \in \Omega_0} v_s u_s^2}\right) + \frac{1}{n} \sum_{j=1}^m f_j \log(w_j) \\ &= \log\left(\frac{1}{n-p} \sum_{i=1}^{n-p} u_i^2\right) - \log\left(\frac{\sum_{j=0}^m f'_j w_j U_j}{n-p} + \frac{\sum_{s \in \Omega_0} v_s u_s^2}{n-p}\right) + \frac{1}{n} \sum_{j=1}^m f_j \log(w_j) \end{aligned}$$

Looking at the first term, let's note that $u_i \sim N(0, 1)$ for $i = 1, \dots, n-p$. This allows us to rewrite the first term and note that:

$$\begin{aligned} \frac{1}{n-p} \sum_{i=1}^{n-p} u_i^2 &= \frac{1}{n-p} \sum_{i=1}^{n-p} (u_i - 0)^2 = S_u \rightarrow^p \sigma_u = 1 \\ \frac{1}{n-p} \sum_{i=1}^{n-p} u_i^2 &\rightarrow^p 1 \end{aligned}$$

by the weak law of large numbers.

$$\Rightarrow \log\left(\frac{1}{n-p} \sum_{i=1}^{n-p} u_i^2\right) \rightarrow^p \log(1) = 0$$

Looking at the second term, we note that $U_j = \frac{1}{f'_j} \sum_{i \in \Omega_j} u_i^2$ for $j = 0, \dots, m$. Additionally, we note here that $f'_j = |\Omega_j|$ which is the count of the times that ϕ_j is replicated in non-zero eigenvalues. For the purposes of this, it is sufficient to say that $|\Omega_j|$ is a count.

We can rewrite U_j as:

$$U_j = \frac{1}{|\Omega_j|} \sum_{i \in \Omega_j} (u_i - 0)^2 \rightarrow^p 1$$

since $u_i \sim N(0, 1)$ as previously stated.

Additionally, this implies that as $U_j \rightarrow^p 1$:

$$\Rightarrow \sum_{s \in \Omega_0} \frac{v_s u_s^2}{n-p} \rightarrow^p 0$$

under the null hypothesis.

This is since $v_s = 1$ under the null hypothesis:

$$v_s = \frac{1}{1 + \lambda \rho_s} = \frac{1}{1 + 0} = 1$$

Additionally, as long as $\sum_{s \in \Omega_0} u_s^2$ is finite, then $\sum_{s \in \Omega_0} \frac{v_s u_s^2}{n-p} = \sum_{s \in \Omega_0} \frac{u_s^2}{n-p} \rightarrow 0$ as $(n-p) \rightarrow \infty$.

Noted in the paper:

$$\begin{aligned} \frac{f'_j}{n-p} &\rightarrow q_j \\ \frac{f_j}{n} &\rightarrow q_j \end{aligned}$$

The suggests that the second term of the equation:

$$-\log \left(\frac{\sum_{j=0}^m f'_j w_j U_j}{n-p} + \frac{\sum_{s \in \Omega_0} v_s u_s^2}{n-p} \right) \rightarrow^p -\log \left(\sum_{j=0}^m q_j w_j + 0 \right) = -\log \left(\sum_{j=0}^m q_j w_j \right)$$

The third element of the equation converges:

$$\sum_{j=1}^m \frac{f_j}{n} \log(w_j) \rightarrow^p \sum_{j=1}^m q_j \log(w_j)$$

Let's now look $G_n(\lambda)$. Let's start by rewriting $G_n(\lambda)$:

$$\begin{aligned} G_n(\lambda) &= \log \left(\frac{\sum_{i=1}^{n-p} u_i^2}{\sum_{j=0}^m f'_j w_j U_j + \sum_{s \in \Omega_0} v_s u_s^2} \right) + \frac{1}{n-p} \sum_{j=1}^m f'_j \log(w_j) + \frac{1}{n-p} \sum_{s \in \Omega_0} \log(v_s) \\ &= \log \left(\frac{n-p}{n-p} \cdot \frac{\sum_{i=1}^{n-p} u_i^2}{\sum_{j=0}^m f'_j w_j U_j + \sum_{s \in \Omega_0} v_s u_s^2} \right) + \frac{1}{n-p} \sum_{j=1}^m f'_j \log(w_j) + \frac{1}{n-p} \sum_{s \in \Omega_0} \log(v_s) \\ &= \log \left(\frac{1}{n-p} \sum_{j=1}^{n-p} u_j^2 \right) - \log \left(\frac{1}{n-p} \left(\sum_{j=0}^m f'_j w_j U_j + \sum_{s \in \Omega_0} v_s u_s^2 \right) \right) + \sum_{j=1}^m \frac{f'_j}{n-p} \log(w_j) + \frac{\sum_{s \in \Omega_0} \log(v_s)}{n-p} \end{aligned}$$

In the previous proof of $F_n(\lambda)$ we showed the convergence of the first and second elements of $G_n(\lambda)$.

For the third element, recall that $\frac{f'_j}{n-p} \rightarrow q_j$ which implies that $\sum_{j=1}^m \frac{f'_j}{n-p} \log(w_j) \rightarrow^p \sum_{j=1}^m q_j \log(w_j)$ which has been shown that the supremum of this third element converges in probability to 0.

The final element of this equation converges to 0, since under the null hypothesis $v_s \rightarrow 1$ as previously shown.

Therefore by the uniform convergence theorem, $\sup_{\lambda \in [-\delta, T]} |G_n(\lambda) - F_0(\lambda)| \rightarrow^p 0$. $G_n(\lambda)$ therefore converges uniformly to $F_0(\lambda)$ in probability over $\lambda \in [-\delta, T]$.

ii) Recall that $F_0(\lambda) = \sum_{j=0}^m q_j \log(w_j) - \log\left(\sum_{j=0}^m q_j w_j\right)$.

The proof that the authors used involved Jensen's inequality to show that $F_0(\lambda)$ achieves its unique maximum at $\lambda = 0$. I was confused by this and will include my thought process in coming to the conclusion that I did.

When evaluating under the null hypothesis at $\lambda = 0$, we really only need to worry about the second term $-\log(\sum_{j=0}^m q_j w_j)$. Additionally, when considering with the constrain of the bounded parameter space $\lambda \geq 0$, we can see that $F_0(\lambda) \leq 0$ when $\lambda \geq 0$. We also note that q_j is a proportion such that $\sum_{j=0}^m q_j \leq 1$.

Under these constraints.

$$\begin{aligned} F_0(0) &= -\log\left(\sum_{j=0}^m q_j\right) \mid \sum_{j=0}^m q_j \leq 1 \\ \Rightarrow F_0(0) &\geq 0 \text{ when } \lambda = 0 \end{aligned}$$

Since λ can only be non-negative with the way it has been defined in this paper:

$$\begin{aligned} \left|\sum_{j=0}^m q_j \log\left(\frac{1}{1 + \lambda \phi_j}\right)\right| &> \left|\log\left(\sum_{j=0}^m q_j \frac{1}{1 + \lambda \phi_j}\right)\right| \\ &\Rightarrow F_0(\lambda) \leq 0 \end{aligned}$$

Which suggests that $\lambda = 0$ is the unique maximum of $F_0(\lambda)$.

iii) Since $\lambda = 0$ has been established as the unique global maximum, we can reference theorem 5.7 of *Asymptotic Statistics* (1998) by A. van der Vaart, which states that if there is a unique global estimator with the properties we have already established, then $\hat{\theta} \rightarrow^p \theta_0$. In this case, both the MLE $\hat{\lambda}_n$ and the REML $\hat{\lambda}_n^r$ have the unique global maximum at $\lambda = 0$ and would both converge in probability to 0 according to this theorem.

3.2 Theorem 1

Suppose the $\sup_{s \in \Omega_0} \leq B$ are bounded and $p = o(\sqrt{n-p})$. Then, under the null hypothesis (i) $\sqrt{n-p}\hat{\lambda}_n \rightarrow^p \frac{1}{2}1_{\{0\}} + \frac{1}{2}N(0, \sigma_\lambda^2)^+$, as $n \rightarrow \infty$; (ii) $\sqrt{n-p}\hat{\lambda}_n^r \rightarrow^D \frac{1}{2}1_{\{0\}} + \frac{1}{2}N(0, \sigma_\lambda^2)^+$, as $(n-p) \rightarrow \infty$; where $\sigma_\lambda^2 = \frac{2}{V(Z_\phi)}$ and $N(0, \sigma_\lambda^2)^+ = \max(0, N(0, \sigma_\lambda^2))$ is the left truncated distribution of $N(0, \sigma_\lambda^2)$ at 0.

As previously discussed in Lemma 3, $\hat{\lambda}_n$ and $\hat{\lambda}_n^r$ are the global maximizers under the null over the interval $[-\delta, T]$.

i)

Let's think of MLE $\hat{\lambda}_n$ as the left truncated at 0. Such that:

$$\hat{\lambda}_n = \begin{cases} \tilde{\lambda}_n & \tilde{\lambda}_n > 0 \\ 0 & \tilde{\lambda}_n \leq 0 \end{cases}$$

When n is large enough, note that $F'_n(\tilde{\lambda}_n) = 0$.

If we take the Taylor series expansion of F'_n around 0 we get:

$$0 = F'_n(0) + \tilde{\lambda}_n F''_n(0) + \frac{1}{2} \tilde{\lambda}_n^2 F'''_n(c_n \tilde{\lambda}_n)$$

where $c \in [0, 1]$ and we define $c_n \tilde{\lambda}_n = \eta_n$

We can rewrite the Taylor expansion.

$$\begin{aligned} -F'_n(0) &= \tilde{\lambda}_n F''_n(0) + \frac{1}{2} \tilde{\lambda}_n^2 F'''_n(\eta_n) \\ \Rightarrow \tilde{\lambda}_n (F''_n(0) + \frac{1}{2} \tilde{\lambda}_n F'''_n(\eta_n)) &= -F'_n(0) \\ \Rightarrow \sqrt{n-p} \tilde{\lambda}_n &= \frac{\sqrt{n-p} F'_n(0)}{-F''_n(0) - \frac{1}{2} \tilde{\lambda}_n F'''_n(\eta_n)} \end{aligned}$$

We can find $F'_n(0)$ and $F''_n(0)$ by taking the derivatives with respect to λ and evaluating at $\lambda = 0$.

$$\begin{aligned} F'_n(0) &= \frac{\partial}{\partial \lambda} \left[\frac{\sum_{i=1}^{n-p} u^2}{\sum_{j=0}^m f_j \frac{1}{1+\lambda \phi_j} U_j + \sum_{s \in \Omega_0} \frac{1}{1+\lambda \rho_s} u_s^2} + \frac{1}{n} \sum_{j=1}^m f_j \log \left(\frac{1}{1+\lambda \phi_j} \right) \right]_{\lambda=0} \\ &= \frac{\sum_{j=0}^m \frac{f_j}{n-p} \phi_j U_j + \sum_{s \in \Omega_0} \frac{\rho_s u_s^2}{n-p}}{\sum_{j=0}^m \frac{f'_j}{n-p} U_j + \sum_{s \in \Omega_0} \frac{u_s^2}{n-p}} - \sum_{j=1}^m \frac{f_j}{n} \phi_j \\ F''_n(0) &= \left(\frac{\sum_{j=0}^m \frac{f_j}{n-p} \phi_j U_j + \sum_{s \in \Omega_0} \frac{\rho_s u_s^2}{n-p}}{\sum_{j=0}^m \frac{f'_j}{n-p} U_j + \sum_{s \in \Omega_0} \frac{u_s^2}{n-p}} \right)^2 - \frac{2 \sum_{j=0}^m \frac{f'_j}{n-p} \phi_j^2 U_j + 2 \sum_{s \in \Omega_0} \frac{\rho_s^2 u_s^2}{n-p}}{\sum_{j=0}^m \frac{f'_j}{n-p} U_j + \sum_{s \in \Omega_0} \frac{u_s^2}{n-p}} + \sum_{j=1}^m \frac{f_j}{n} \phi_j^2 \end{aligned}$$

Let $\bar{\rho}_n = \sum_{j=0}^m \frac{f'_j}{n-p} \phi_j$ and $\bar{\zeta}_n = \sum_{j=0}^m \frac{f_j}{n} \phi_j$.

We can now find what the different parts of the equation converge to.

$$\begin{aligned} \sqrt{n-p} F'_n(0) &= \sqrt{n-p} \left(\frac{\sum_{j=0}^m \frac{f'_j}{n-p} \phi_j U_j + O\left(\frac{p}{n-p}\right)}{\sum_{j=0}^m \frac{f'_j}{n-p} U_j + O\left(\frac{p}{n-p}\right)} - \bar{\zeta}_n \right) \\ &= \frac{\sum_{j=0}^m \sqrt{\frac{f'_j}{n-p}} (\phi_j - \bar{\zeta}_n) \sqrt{f'_j} (U_j - 1) + \sqrt{n-p} (\bar{\rho}_n - \bar{\zeta}_n) + O\left(\frac{p}{\sqrt{n-p}}\right)}{\sum_{j=0}^m \frac{f'_j}{n-p} U_j + O\left(\frac{p}{n-p}\right)} \end{aligned}$$

Here we note that $\bar{\rho}_n \rightarrow \sum_{j=0}^m q_j \phi_j = \bar{\phi}$ and $\bar{\zeta}_n \rightarrow \bar{\phi}$.

Additionally, we note the following:

$$\begin{aligned} \sqrt{f'_j(U_j - 1)} &\rightarrow^D z_i = N(0, 2) \\ \sum_{j=0}^m \frac{f'_j}{n-p} U_j + O\left(\frac{p}{n-p}\right) &\rightarrow^p 1 \end{aligned}$$

Also, let $u^* = \lim_{n \rightarrow +\infty} \sqrt{n-p}(\bar{\rho}_n - \bar{\zeta}_n) = \lim_{n \rightarrow +\infty} u_n = 0$. Recall that in the paper $V(Z_\phi) = \sum_{j=0}^m q_j(\phi_j - 1)^2$

Under those conditions and by Slutsky's theorem

$$\sqrt{n-p} F'_n(0) \rightarrow^D N(u^*, 2V(Z_\phi))$$

For the second part of the equation, note that:

$$F''_n(0) \rightarrow^p F''_n(\lambda_0) = -V(Z_\phi)$$

We note that $F'''_n(\eta_n)$ is finite and the $\hat{\lambda}_n \rightarrow^p 0$ as defined by Lemma 3. This implies that $\hat{\lambda}_n F'''_n(\eta_n) \rightarrow^p 0$.

This lets us say that

$$\sqrt{n-p} \tilde{\lambda}_n \rightarrow^D N\left(0, \frac{2}{V(Z_\phi)}\right)$$

And since this hypothesis test is bounded on the edge of the parameter space, we can say that:

$$\sqrt{n-p} \hat{\lambda}_n \rightarrow^D \frac{1}{2}(1) + \frac{1}{2}N\left(0, \frac{2}{V(Z_\phi)}\right)$$

ii)

This proof is pretty similar to part i), with a few minor changes since we will be working with $G_n(\lambda)$. We essentially follow the same steps.

When doing the Taylor's expansion around 0 and solving for $\tilde{\lambda}_n^r$ we get:

$$\sqrt{n-p} \tilde{\lambda}_n^r = \frac{\sqrt{n-p} G'_n(0)}{-G''_n(0) - \frac{1}{2} \tilde{\lambda}_n^r G''_n(\eta_n)} - \frac{\sum_{s \in \Omega_0} \rho_s v_s^2}{n-p}$$

Taking the derivatives of $G_n(\lambda)$ and evaluating at 0 to find $G'_n(0)$ and $G''_n(0)$:

$$\begin{aligned} G'_n(0) &= \frac{\sum_{j=0}^m \frac{f'_j}{n-p} \phi_j U_j + \frac{1}{n-p} \sum_{s \in \Omega_0} \rho_s u_s^2}{\sum_{j=0}^m \frac{f'_j}{n-p} U_j + \frac{1}{n-p} \sum_{s \in \Omega_0} u_s^2} - \sum_{j=1}^m \frac{f'_j}{n-p} \phi_j - \sum_{s \in \Omega_0} \frac{\rho_s}{n-p} \\ G''_n(0) &= \left(\frac{\sum_{j=0}^m \frac{f'_j}{n-p} \phi_j U_j + \frac{1}{n-p} \sum_{s \in \Omega_0} \rho_s u_s^2}{\sum_{j=0}^m \frac{f'_j}{n-p} U_j + \frac{1}{n-p} \sum_{s \in \Omega_0} u_s^2} \right)^2 - \frac{2 \sum_{j=0}^m \frac{f'_j}{n-p} \phi_j U_j + \frac{1}{n-p} \sum_{s \in \Omega_0} \rho_s^2 u_s^2}{\sum_{j=0}^m \frac{f'_j}{n-p} U_j + \frac{1}{n-p} \sum_{s \in \Omega_0} u_s^2} + \sum_{j=1}^m \frac{f'_j}{n-p} \phi_j^2 + \sum_{s \in \Omega_0} \frac{\rho_s^2}{n-p} \end{aligned}$$

Under Lemma 1 of the paper, $\sum_{s \in \Omega_0} \rho_s = O(p)$. Also define $\bar{\phi} = \sum_{j=0}^m q_j \phi_j$ and $\bar{\rho}_n = \sum_{j=0}^m \frac{f'_j}{n-p} \phi_j$. This implies that:

$$\sqrt{n-p} G'_n(0) = \frac{\sum_{j=0}^m \sqrt{\frac{f'_j}{n-p}} (\phi_j - \bar{\rho}_n) \sqrt{f'_j} (U_j - 1) + O\left(\frac{1}{\sqrt{n-p}}\right)}{\sum_{j=0}^m \frac{f'_j}{n-p} U_j + O\left(\frac{p}{n-p}\right)} + O\left(\frac{p}{\sqrt{n-p}}\right)$$

We note the following:

$$\begin{aligned} \sqrt{f'_j} (U_j - 1) &\rightarrow^D z_j \sim N(0, 2) \\ \sum_{j=0}^m \frac{f'_j}{n-p} U_j + O\left(\frac{p}{n-p}\right) &\rightarrow^p 1 \\ \Rightarrow \sqrt{n-p} G'_n(0) &\rightarrow^D \sum_{j=0}^m \sqrt{q_j} (\phi_j - \bar{\phi}) z_j \end{aligned}$$

Additionally note that:

$$G''_n(0) \rightarrow^p \phi^2 - \sum_{j=1}^m q_j \phi_j^2 = -V(Z_\phi)$$

And, since $G'''_n(\eta_n)$ is finite we can say that $\tilde{\lambda}_n^r G'''_n(\eta_n) \rightarrow^p 0$

By Slutsky's:

$$\begin{aligned} \sqrt{n-p} \tilde{\lambda}_n^r &\rightarrow^D N(0, \frac{2}{V(Z_\phi)}) \\ \sqrt{n-p} \hat{\lambda}_n^r &\rightarrow^D \frac{1}{2} 1 + \frac{1}{2} N(0, \frac{2}{V(Z_\phi)}) \end{aligned}$$

3.3 Corollary 1

Under the same assumptions as in Theorem 1, i) $LRT_n \rightarrow^D \frac{1}{2} 1_0 + \frac{1}{2} \chi_1^2$, as $n \rightarrow \infty$; ii) $RLRT_n \rightarrow^D \frac{1}{2} 1_{\{0\}} + \frac{1}{2} \chi_1^2$, as $(n-p) \rightarrow \infty$.

i)

If we take a Taylor Expansion around $F_n(\tilde{\lambda}_n)$ gives us:

$$F_n(\tilde{\lambda}_n) = \tilde{\lambda}_n (F'_n(0)) + \frac{1}{2} \tilde{\lambda}_n F''_n(0) + \frac{1}{6} \tilde{\lambda}_n^2 F'''_n(\tilde{\eta}_n)$$

where $\tilde{\eta}_n = \tilde{c}_n \tilde{\lambda}_n$ and $\tilde{c}_n \in [0, 1]$.

Recall that according to the proof for Theorem 1, $F'_n(0)$ can be written as:

$$F'_n(0) = -\tilde{\lambda}_n F''_n(0) - \frac{1}{2} \tilde{\lambda}_n^2 F'''_n(\eta_n)$$

We can combine the two equations and get:

$$(n-p)F_n(\tilde{\lambda}_n) = -\frac{1}{2}(n-p)\tilde{\lambda}_n^2 F_n''(0) + (n-p)\tilde{\lambda}_n^3 \left[\frac{1}{6}F_n'''(\tilde{\eta}_n) - \frac{1}{2}F_n'''(\eta_n) \right]$$

We note here that $F_n'''(\eta_n)$ and $F_n'''(\tilde{\eta}_n)$ are both finite and $\tilde{\lambda}_n \rightarrow^p 0$ which allows us to rewrite the equation as:

$$(n-p)F_n(\tilde{\lambda}_n) = -\frac{F_n''(0)}{2}(\sqrt{n-p}\tilde{\lambda}_n)^2$$

Recall from the previous proof that $-F_n''(0) \rightarrow^p V(Z_\phi)$ and $\sqrt{n-p}\tilde{\lambda}_n \rightarrow^D N\left(0, \frac{2}{V(Z_\phi)}\right)$.

Thus, when $\tilde{\lambda}_n > 0$, $(n-p)F_n(\tilde{\lambda}_n) \rightarrow^D \chi_1^2$ and

$$LRT_n \rightarrow^D \frac{1}{2}1_{\{0\}} + \frac{1}{2}\chi_1^2$$

as $n \rightarrow \infty$.

ii)

For the RLRT, we start with doing a Taylor's expansion of G_n around 0:

$$G_n(\tilde{\lambda}_n^r) = \tilde{\lambda}_n^r(G_n'(0) + \frac{1}{2}\tilde{\lambda}_n^r G_n''(0) + \frac{1}{6}(\tilde{\lambda}_n^r)G_n'''(\tilde{\eta}_n))$$

Where $\tilde{\eta}_n = \tilde{c}_n \tilde{\lambda}_n$ and $\tilde{c}_n \in [0, 1]$.

Note that from the Theorem 1 proof we can write $G - n'(0)$ as:

$$G_n'(0) = -\tilde{\lambda}_n^r G_n''(0) - \frac{1}{2}(\tilde{\lambda}_n^r)^2 G_n'''(\eta_n)$$

We can use this equation to get:

$$(n-p)G_n(\tilde{\lambda}_n^r) = -\frac{1}{2}(n-p)(\tilde{\lambda}_n^r)^2 G_n''(0) + (n-p)(\tilde{\lambda}_n^r)^3 \left[\frac{1}{6}G_n'''(\tilde{\eta}_n) - \frac{1}{2}G_n'''(\eta_n) \right]$$

Since $G_n'''(\tilde{\eta}_n)$ and $G_n'''(\eta)$ are finite and $(\tilde{\lambda}_n^r) \rightarrow^p 0$ we can rewrite this as:

$$(n-p)G_n(\tilde{\lambda}_n^r) = -\frac{1}{2}(n-p)(\tilde{\lambda}_n^r)^2 G_n''(0)$$

Which can again be rewritten as:

$$(n-p)G_n(\tilde{\lambda}_n^r) = \left(-\frac{G_n''(0)}{V(Z_\phi)} \right) \left(\frac{V(Z_\phi)(n-p)}{2} (\tilde{\lambda}_n^r)^2 \right)$$

From the proof on Theorem 1, we can note that $-G_n''(0) \rightarrow^p V(Z_\phi)$. The second term in the equation, $\left(\frac{V(Z_\phi)(n-p)}{2} (\tilde{\lambda}_n^r)^2 \right) \rightarrow \chi_1^2$.

Therefore, when $\tilde{\lambda}_n^r > 0$:

$$(n-p)G_n(\tilde{\lambda}_n^r) \rightarrow^D \chi_1^2$$

$$\Rightarrow RLRT \rightarrow^D \frac{1}{2}1_{\{0\}} + \frac{1}{2}\chi_1^2$$

4 Simulation Study

4.1 Balanced one-way analysis of variance

4.1.1 Algorithm to simulate the null finite sample distribution of LRT_n

Step 1: define a grid $0 = \lambda_1 < \lambda_2 < \dots < \lambda_m$ of possible values for λ .

Step 2: simulate K independent χ_1^2 random variables w_1^2, \dots, w_K^2 . Set $S_K = \sum_{s=1}^K w_s^2$.

Step 3: independently of step 1, simulate $X_{n,K,p} = \sum_{s=K+1}^{n-p} w_s^2$ with a χ_{n-p-K}^2 -distribution.

Step 4: for every grid point λ_i compute

$$N_n(\lambda_i) = \sum_{s=1}^K \frac{\lambda_i \mu_{s,n}}{1 + \lambda_i \mu_{s,n}} w_s^2$$

$$D_n(\lambda_i) = \sum_{s=1}^K \frac{w_s^2}{1 + \lambda_i \mu_{s,n}} w_s^2 + X_{n,K,p}$$

Step 5: determine λ_{max} which maximizes $f_n(\lambda_i)$ over $\lambda_1, \dots, \lambda_m$.

Step 6: compute

$$LRT_n = f_n(\lambda_{max})$$

Step 7: repeat steps 2-6 until the desired number of simulations is achieved.

4.2 Familial

```
pkgs <- c('dplyr',
          'Matrix',
          'tictoc',
          'RSpectra',
          'parallel',
          'lme4qtl',
          'lme4',
          'data.table',
          'pbmcapply')
invisible(sapply(pkgs, library, character.only = TRUE))
`%+%` <- function(...) paste0(...)
##
## Set global parameters #####
```



```

### Name of simulation run/setting ###
name <- 'test'

### Family types ###
a_1 <- Matrix(data = c(1.0, 0.0, 0.5,
                      0.0, 1.0, 0.5,
                      0.5, 0.5, 1.0),
              nrow = 3,
              ncol = 3,
              byrow = TRUE)
a_2 <- Matrix(data = c(1.0, 0.0, 0.5, 0.5,
                      0.0, 1.0, 0.5, 0.5,
                      0.5, 0.5, 1.0, 0.5,
                      0.5, 0.5, 0.5, 1.0),
              nrow = 4,
              ncol = 4,
              byrow = TRUE)
a_3 <- Matrix(data = c(1.0, 0.0, 0.5, 0.5, 0.5,
                      0.0, 1.0, 0.5, 0.5, 0.5,
                      0.5, 0.5, 1.0, 0.5, 0.5,
                      0.5, 0.5, 0.5, 1.0, 0.5,
                      0.5, 0.5, 0.5, 0.5, 1.0),
              nrow = 5,
              ncol = 5,
              byrow = TRUE)
a_4 <- Matrix(data = c(1.0, 0.0, 0.5, 0.5, 0.5, 0.5,
                      0.0, 1.0, 0.5, 0.5, 0.5, 0.5,
                      0.5, 0.5, 1.0, 0.5, 0.5, 0.5,
                      0.5, 0.5, 0.5, 1.0, 0.5, 0.5,
                      0.5, 0.5, 0.5, 0.5, 1.0, 0.5,
                      0.5, 0.5, 0.5, 0.5, 0.5, 1.0),
              nrow = 6,
              ncol = 6,
              byrow = TRUE)
as <- lapply(list(a_1,a_2,a_3,a_4),
             function(x) t(chol(x)))

### Number of random normal fixed effects ###
p <- 1

### Fixed effect coefficients and names ###
beta <- rep(1, p)
xnames <- 'x' %>% 1:p

### Weights for family types ###
ws <- c(0.2,0.3,0.3,0.2)

### Number of families to choose: (10,50,100,200,400) ###
K <- 100

### Error variance ###

```

```

s2E <- 1

### Heritability: (0.0, 0.1, 0.2, 0.5) ###
h2 <- 0.0

### Alphas (significance level) to consider ###
alphas <- c(0.05, 0.01)

### Number of samples for exact distribution ###
samps_exact <- 10000

### How many cores to use to calc exact distribution ###
cores_exact <- 1

### Number of simulations ###
nsim <- 100

### Number of cores for simulation ###
cores_sim <- 1

### Statistics to calculate ###
statistics <- c('LRT', 'RLRT')

### Fit linear model ###
##-- X does NOT include the intercept, one is automatically added
##-- corr MUST have row and column names for subject ID (same order as Y and X)
fit_lmm <- function(Y,
                    X,
                    corr = NULL,
                    REML) {
  if(!is.null(corr)) {
    d <- data.frame('y' = as.matrix(Y),
                    as.matrix(X),
                    id = colnames(corr))
    model <- relmatLmer(y ~ . - id + (1|id),
                      d,
                      relmat = list(id = corr),
                      REML = REML)
    beta <- getME(model, name = "fixef")
    beta <- beta[order(names(beta))]

    foo <- as.data.frame(VarCorr(model))[, c("var1", "vcov")] %>%
      mutate(var1 = case_when(is.na(var1) ~ "s2E",
                             TRUE ~ 's2A'))
    vcs <- setNames(foo$vcov, foo$var1)
    vcs <- vcs[order(names(vcs))]
    ll_REML <- logLik(model, REML = TRUE)[1]
    ll_ML <- logLik(model, REML = FALSE)[1]
    rss <- unname(getME(model, "devcomp")$cmp["pwrss"])
    list("type" = "lme4",
         "model" = model,

```

```

    "beta" = beta,
    "vcs" = vcs,
    "ll_REML" = ll_REML,
    "ll_ML" = ll_ML,
    "rss" = rss,
    "method" = ifelse(REML, "REML", "ML"))
} else {
  d <- data.frame('y' = as.matrix(Y),
                  as.matrix(X))
  model <- lm(y ~ .,
             d)
  X <- model.matrix(model)
  beta <- model$coefficients
  r <- Y - X %*% beta
  rss <- {t(r) %*% r}[1, 1]
  if (!REML) {
    vcs <- c("s2E" = rss / nrow(X))
  } else {
    vcs <- c("s2E" = rss / (nrow(X) - ncol(X)))
  }
  lnum <- log(2 * pi * rss)
  d <- 0
  dd <- det(t(X) %*% X)
  ll_REML <- logLik(model, REML = TRUE)[1]
  ll_ML <- logLik(model, REML = FALSE)[1]
  beta <- beta[order(names(beta))]
  list("type" = "lm",
       "model" = model,
       "beta" = beta,
       "vcs" = vcs,
       "rss" = rss,
       "ll_REML" = ll_REML,
       "ll_ML" = ll_ML,
       "method" = ifelse(REML, "REML", "ML"))
}
}

### Get critical values of 50:50 0 and Chi Square 1 ####
qmix <- function(q) {
  if (q < 0.5) {
    0
  } else {
    qchisq(2*(q - 0.5), 1)
  }
}
qmix <- Vectorize(qmix)

### Get time ####
get_time <- function() {
  format(Sys.time(), "%Y_%m_%d %I_%M_%S_%p")
}

```

```

## Create functions for simulation ####

### Generate what variance components should be ###
gen_vc <- function(h2, s2E) {
  s2A <- h2 / (1 - h2) * s2E
  list(h2 = h2, s2A = s2A, s2E = s2E)
}

### Generate data ####
generate_data <- function(K,
                           p,
                           ws,
                           vc,
                           beta) {
  fam_types <- sample(1:length(as),
                     K,
                     TRUE,
                     ws)
  a <- lapply(fam_types,
             function(x) as[[x]])
  A <- bdiag(a)
  AAt <- tcrossprod(A)
  n <- nrow(A)
  X <- Matrix(rbinom(n*p,1,0.5),
             ncol = p)
  P <- diag(n) - tcrossprod(qr.Q(qr(X)))
  rho <- eigen(t(A) %*% P %*% A,
              symmetric = TRUE,
              only.values = TRUE)$values

  phi <- eigen(AAt,
              symmetric = TRUE,
              only.values = TRUE)$values
  a <- rnorm(n, sd = sqrt(vc$s2A))
  e <- rnorm(n, sd = sqrt(vc$s2E))
  Y <- {X%*%beta + A %*% a + e}[, 1]
  colnames(X) <- xnames
  ids <- 'id'%*%{1:n}
  colnames(AAt) <- ids
  rownames(AAt) <- ids
  list(A = A,
       AAt = AAt,
       X = X,
       P = P,
       rho = rho,
       phi = phi,
       Y = Y,
       n = n,
       p = p)
}

```

```

### Get exact distributions ###
#- just one
get_exact_dist_one <- function(data,
                               statistics) {

  out <- list()
  ##-- common things
  u2 <- rnorm(data$n)^2
  logRSS0 <- log(sum(u2))
  ##-- LRT
  if('LRT' %in% statistics) {
    f_n <- function(lambda) {
      onepluslambdarho <- 1 + lambda * data$rho
      onepluslambdaphi <- 1 + lambda * data$phi
      data$n * log(sum(u2/onepluslambdarho)) + sum(log(onepluslambdaphi))
    }
    out['LRT'] <- data$n * logRSS0 -
      optim(par = 0.5,
            fn = f_n,
            lower = 0,
            method = "L-BFGS-B")$value
  }

  ##-- RLRT
  if ('RLRT' %in% statistics) {
    g_n <- function(lambda) {
      onepluslambdarho <- 1 + lambda * data$rho
      (data$n - data$p) * log(sum(u2/onepluslambdarho)) + sum(log(onepluslambdarho))
    }
    out["RLRT"] <- (data$n - data$p) * logRSS0 -
      optim(par = 0.5,
            fn = g_n,
            lower = 0,
            method = "L-BFGS-B")$value
  }
  #- Get many
  get_exact_dist <- function(data,
                              statistics,
                              samps_exact,
                              cores_exact) {
    bind_rows(mclapply(1:samps_exact,
                       function(x) {
                         o <- get_exact_dist_one(data = data,
                                                    statistics = statistics)
                         o$sim <- x
                         o
                       },
                       mc.cores = cores_exact))
  }

  ### Calculate test statistics ###
  get_statistics <- function(data,

```

```

        statistics) {
stats <- list()
ests <- list()
##-- LRT
if('LRT' %in% statistics) {
  ###--- Fit alternative model

  ml_model_A <- fit_lmm(Y = data$Y,
                        X = data$X,
                        corr = data$AAat,
                        REML = FALSE)
  ests[['ml_A']] <- bind_rows(tibble(var = names(ml_model_A$vcs),
                                    value = ml_model_A$vcs),
                             tibble(var = 'beta_' %>% names(ml_model_A$beta),
                                    value = ml_model_A$beta)) %>%

    mutate(method = 'ml',
           hypoth = 'hA')

  ###--- Fit null model
  ml_model_0 <- fit_lmm(Y = data$Y,
                        X = data$X,
                        REML = FALSE)
  ests[['ml_0']] <- bind_rows(tibble(var = names(ml_model_0$vcs),
                                    value = ml_model_0$vcs),
                             tibble(var = 'beta_' %>% names(ml_model_0$beta),
                                    value = ml_model_0$beta)) %>%

    mutate(method = 'ml',
           hypoth = 'h0') %>%
  bind_rows(tibble(hypoth = rep(c('h0', 'hA'), each = 2),
                    var = rep(c('ll', 'rll'), 2),
                    method = 'ml',
                    value = c(ml_model_0$ll_ML,
                              ml_model_0$ll_REML,
                              ml_model_A$ll_ML,
                              ml_model_A$ll_REML)))
  stats["LRT"] <- 2 * (ml_model_A$ll_ML - ml_model_0$ll_ML)
}

##-- Things that depend on restricted likelihood (could be optimized)
if(any(c('RLRT', 'Fstat', 'Q', 'Qtilde') %in% statistics)) {
  ###--- Fit alternative model
  rml_model_A <- fit_lmm(Y = data$Y,
                        X = data$X,
                        corr = data$AAat,
                        REML = TRUE)
  ests[['rml_A']] <- bind_rows(tibble(var = names(rml_model_A$vcs),
                                    value = rml_model_A$vcs),
                             tibble(var = 'beta_' %>% names(rml_model_A$beta),
                                    value = rml_model_A$beta)) %>%

    mutate(method = 'rml',
           hypoth = 'hA')

```

```

###--- Fit null model
rml_model_0 <- fit_lmm(Y = data$Y,
                      X = data$X,
                      REML = TRUE)
ests[['rml_0']] <- bind_rows(tibble(var = names(rml_model_0$vcs),
                                   value = rml_model_0$vcs),
                           tibble(var = 'beta_' %>% names(rml_model_0$beta),
                                   value = rml_model_0$beta)) %>%

mutate(method = 'rml',
       hypoth = 'h0') %>%
bind_rows(tibble(hypoth = rep(c('h0', 'hA'), each = 2),
                 var = rep(c('ll', 'rll'), 2),
                 method = 'rml',
                 value = c(rml_model_0$ll_ML,
                           rml_model_0$ll_REML,
                           rml_model_A$ll_ML,
                           rml_model_A$ll_REML)))

###--- Calculate some things that are re-rused (could be optimized)
r <- Matrix(rml_model_0$model$residuals, ncol = 1)
Qs_num <- (t(r) %*% data$Aat %*% r)
###--- RLRT
if ("RLRT" %in% statistics) {
  stats["RLRT"] <- 2 * (rml_model_A$ll_REML - rml_model_0$ll_REML)
}
}

###--- output
list('stats' = tibble(Statistic = names(stats),
                     value = unlist(stats)),
     'ests' = bind_rows(ests))
}

one_simulation <- function(sim,
                          K,
                          P,
                          ws,
                          vc,
                          beta,
                          statistics,
                          samps_exact,
                          cores_exact,
                          crit_asym,
                          alphas) {
  ###--- Generate data
  data <- generate_data(K = K,
                       p = p,
                       ws = ws,
                       vc = vc,
                       beta = beta)

  ###--- Get exact distribution
  dist_exact <- get_exact_dist(data = data,
                              statistics = statistics,

```

```

                                samps_exact = samps_exact,
                                cores_exact = cores_exact)

##-- Get critical values
crit <- dist_exact %>%
  group_by(type, statistic) %>%
  summarise(alpha = alphas,
            crit = quantile(value,
                          1 - alphas),
            .groups = 'keep') %>%
  ungroup %>%
  bind_rows(crit_asym)

##-- get statistics
res <- get_statistics(data = data,
                    statistics = statistics)

##-- get reject
res$reject <- res$stats %>%
  left_join(crit,
            by = 'statistic') %>%
  mutate(reject = value > crit) %>%
  select(statistic, type, alpha, reject)

##-- return
res$ests <- res$ests %>%
  mutate(sim = sim)
res$stats <- res$stats %>%
  mutate(sim = sim)
res$reject <- res$reject %>%
  mutate(sim = sim)
##-- record sample size:
res$ssize <- c(data$n, sim)
# %>% mutate(sim = sim)
res
}

sims <- function(name,
                nsim,
                cores_sims,
                K,
                p,
                ws,
                s2E,
                h2,
                beta,
                statistics = statistics,
                samps_exact,
                cores_exact,
                alphas) {
  time <- get_time()
  ##-- Generate VCs
  vc <- gen_vc(h2 = h2,
               s2E = s2E)

```



```

##-- Get critical values
crit_asym <- tibble(type = 'asym',
                    alpha = alphas,
                    crit = qmix(1 - alphas)) %>%
  full_join(tibble(statistic = c('RLRT', 'LRT')),
            by = character()) %>%
  filter(statistic %in% statistics)
##-- do simulation
o <- pbmclapply(1:nsim,
                one_simulation,
                K = K,
                p = p,
                ws = ws,
                vc = vc,
                beta = beta,
                statistics = statistics,
                samps_exact = samps_exact,
                cores_exact = cores_exact,
                crit_asym = crit_asym,
                alphas = alphas,
                mc.cores = cores_sim,
                mc.style = 'txt')
stats <- bind_rows(lapply(o, function(x) x$stats))
rej <- bind_rows(lapply(o, function(x) x$rej))
ests <- bind_rows(lapply(o, function(x) x$ests))
pow <- rej %>%
  group_by(statistic, type, alpha) %>%
  summarize(power = 100*mean(reject)) %>%
  ungroup
##-- record sample size:
ssize <- bind_rows(lapply(o, function(x) x$ssize))
##-- write output
fwrite(stats,
       file = name %++ "_" %++ h2 %++ "_" %++ 'stats' %++ "_" %++ K %++ ".csv")
fwrite(rej,
       file = name %++ "_" %++ h2 %++ "_" %++ 'rej' %++ "_" %++ K %++ ".csv")
fwrite(ests,
       file = name %++ "_" %++ h2 %++ "_" %++ 'ests' %++ "_" %++ K %++ ".csv")
fwrite(pow,
       file = name %++ "_" %++ h2 %++ "_" %++ 'pow' %++ "_" %++ K %++ ".csv")
fwrite(ssize,
       file = name %++ "_" %++ h2 %++ "_" %++ 'ssize' %++ "_" %++ K %++ ".csv")
##-- save settings
settings <- list('name' = name,
                 'nsim' = nsim,
                 'K' = K,
                 'p' = p,
                 'ws' = ws,
                 's2E' = s2E,
                 'h2' = h2,
                 'beta' = beta,

```

```

        'statistics' = statistics,
        'samps_exact' = samps_exact,
        'as' = as)
saveRDS(settings,
        name %++% "_" %++% K %++% ".settings")
}
sims(name = name,
      nsim = nsim,
      cores_sims = cores_sim,
      K = K,
      p = p,
      ws = ws,
      s2E = s2E,
      h2 = h2,
      beta = beta,
      statistics = statistics,
      samps_exact = samps_exact,
      cores_exact = cores_exact,
      alphas = alphas)

```

5 Conclusion