1 Laplace approximation

The coxme function fits the following mixed effects Cox model

$$\lambda(t) = \lambda_0(t)e^{X\beta + Zb}$$

$$b \sim G(0, \Sigma(\theta))$$

where \(\lambda_0\) is an unspecified baseline hazard function, \(X\) and \(Z\) are the design matrices for the fixed and random effects, respectively, \(\beta\) is the vector of fixed-effects coefficients and \(b\) is the vector of random effects coefficients. The random effects distribution \(G\) is modeled as Gaussian with mean zero and a variance matrix \(\Sigma\), which in turn depends a vector of parameters \(\theta\).

The MLE for the variance of the random effects is based on an integrated partial likelihood

$$IPL(\beta, \theta) = \frac{1}{(2\pi)^{q/2}|\Sigma(\theta)|^{1/2}} \int PL(\beta, b) e^{-b'\Sigma^{-1}(\theta)b/2} db$$

where \(q\) is the dimension of the Gaussian density, i.e., the number of random effects. When the variance of the random effect is zero this collapses to the ordinary Cox partial likelihood.

The result of a coxme fit prints out three log-likelihood terms: the fit for a null model where \(\beta=0\) and the variance of the random effect is zero (and therefore \(b=0\)), the log of the integrated value \(IPL(\hat{\beta}, \hat{\theta})\) and the log partial likelihood \(PL(\hat{\beta}, \hat{b})\). (For brevity “log” is not printed in their labels.)

However, the IPL (1) above is not a tractable integral. The key step in its computation is replacement of the log penalized partial likelihood \(LPPL\) with a second order Taylor series about its value at the maximum of the function

$$PL(\beta, b) = e^{\log(PL(\beta,b))} \equiv e^{LPPL(\beta,b)}$$

$$LPPL(\beta, b, \theta) = LPPL(\beta, b) - (1/2)b'^{-1}(\theta)b$$

$$\approx LPPL(\hat{\beta}(\theta), \hat{b}(\theta)) - (1/2)(\beta - \hat{\beta}(\theta), b - \hat{b}(\theta))'H(\beta - \hat{\beta}(\theta), b - \hat{b}(\theta))$$

where the Hessian \(H\) is -1 times the matrix of second derivatives of the LPPL, evaluated at \((\hat{\beta}(\theta), \hat{b}(\theta))\). When \(\theta\) and hence \(A(\theta)\) are fixed, the relevant values of \(\beta\) and \(b\) that maximize the LPPL are easily computed using essentially the same methods as an ordinary Cox model.
For the ML estimate we are only interested in the values at \( \hat{\beta} \) so the last term collapses to 
\((0, \hat{b} - b)'H(0, \hat{b} - b) = (b - \hat{b})'H_{bb}(b - \hat{b})\), where \( H_{bb} \) is the portion of the Hessian corresponding to the random effects. When we replace the body of the integral in (1) with this approximation, then result is an integral that we can solve in closed form. The result is what is printed as the IPL in coxme.

A key question, of course, is whether the result is a good approximation to the IPL. The answer appears to be that it is, if there are a sufficient number of observations that contribute to each random effect. The definition of the word “sufficient” is not completely clear, and the coxme routine includes a option \texttt{refine.n} which does a monte carlo refinement of the final solution, allowing for diagnosis of whether we are in a excellent (often), bad, or intermediate case with respect to the approximation. The remainder of this note gives further details.

### 2 Computation

The central computational strategy for \texttt{coxme} is an outer and an inner loop. The outer loop searches over the parameters \( \theta \) of the variance matrix for a maximum of the IPL. For each trial value of \( \theta \) in this search

1. Calculate \( A(\theta) \) and \( A^{-1}(\theta) \)

2. Solve the penalized Cox model \( LPL(\beta, b) - (1/2)b'A^{-1}b \) to get the solution vector \( (\hat{\beta}, \hat{b}) \), where \( PL \) is the usual Cox partial log-likelihood. The iterative Newton-Raphson solution to this problem is the inner loop.

3. Use the Laplace approximation to compute the log IPL, using the results of step 2.

A necessary component of the solution in step 2 is calculation of \( H \) and its generalized Cholesky decomposition \( H = LDL' \), where \( D \) is diagonal and \( L \) is lower triangular with \( L_{ii} = 1 \). The determinant \( |H| \) is the product of the diagonal elements \( D \). The Laplace approximation in step 3 is particularly convenient for this problem since all the components are already in hand.

### 3 Refining the approximation

#### 3.1 Random treatment effects

As an example case, we first look at a simple simulated data set with random institution and treatment within institution effects.

```r
> library(coxme)
> set.seed(1953) # an auspicious birth year :-)
> mkdata <- function(n, beta=c(.4, .1), sitehaz=c(.5,1.5, 2,1)) {
   nsite <- length(sitehaz)
   site <- rep(1:nsite, each=n)
   trt1 <- rep(0:1, length=n*nsite)
   hazard <- sitehaz[site] + beta[1]*trt1 + beta[2]*trt1 * (site-mean(site))
   stime <- rexp(n*nsite, exp(hazard))
   q80 <- quantile(stime, .8)
```
```r
data.frame(site=site,
    trt = trt1,
    futime= pmin(stime, q80),
    status= ifelse(stime>q80, 0, 1),
    hazard=hazard
}

> trdata <- mkdata(150) #150 enrolled per site
> fit1 <- coxme(Surv(futime, status) ~ trt + (1| site/trt), trdata)
> print(fit1)

Cox mixed-effects model fit by maximum likelihood
Data: trdata
events, n = 480, 600

Iterations= 5 24

Null Integrated Fitted
Log-likelihood -2784.463 -2709.388 -2701.717

Chisq df p AIC BIC
Integrated loglik 150.15 3.00 0 144.15 131.63
Penalized loglik 165.49 4.01 0 157.47 140.72

Model: Surv(futime, status) ~ trt + (1 | site/trt)
Fixed coefficients
  coef exp(coef) se(coef)      z     p
trt   0.2621486  1.29972  0.09288244 2.82 0.0048

Random effects
  Group Variable Std Dev Variance
site/trt (Intercept) 0.0192550177 0.0003707557
site (Intercept) 0.7235930742 0.5235869370

> # Show the true and estimated per-site intercepts
> true <- c(.5, 1.5, 2, 1) - mean(c(.5, 1.5, 2, 1))
> bcoef <- ranef(fit1)[[2]]
> temp <- rbind(true, bcoef)
> dimnames(temp) <- list(c("True", "Estimated"), paste("Site",1:4))
> round(temp,2)

Site 1 Site 2 Site 3 Site 4
True  -0.75  0.25  0.75 -0.25
Estimated -0.86  0.16  0.87 -0.17

The true site hazards have standard deviation \( \sqrt{\text{var}(c(0.5, 1.5, 2, 1))} = 0.65 \), the estimate from the fit is 0.72. In this case the fit has reconstructed the per site intercepts reasonably well.

Figure 1 is a plot of profiles of the likelihood for the four institution effects. We vary \( b \) for each institution while holding all of the other coefficients and the variance fixed. This shows four "slices" through the 12 dimensional LPPL as a function of \( b \). The approximation is not perfect.
Figure 1: The solid lines are profiles of \( PPL(\hat{\beta}, b) \), dashed lines are the Taylor series approximation.

— each LPPL slice is rotated just a little from its quadratic approximation as we move away from the maximum. But remember that we are computing an average of \( \exp(\text{LPPL}) \), so any part of the curves more than 20 units below the max will hardly matter, at least for this small number of dimensions.

Code to draw the figure is below. A `coxph` model with only an offset term is a convenient way to compute the partial likelihood for a fixed model. Also note that random effects are coded using a full contrast matrix. The institution by treatment effects generate 8 random terms \( b_1 \) to \( b_8 \) and the four per-institution intercepts \( b_9 \) to \( b_{12} \). Unlike fixed effects where one of the 4 intercepts would be eliminated due to redundancy (exactly how this is done depends on the contrasts option), the random effects induce two sum constraints

\[
\sum_{1}^{8} b_i = 0 \quad \text{and} \quad \sum_{9}^{12} b_i = 0.
\]

```r
> xx <- seq(-1, 1, length=101) # vary b from -1 to 1
> profile <- matrix(0, nrow=101, ncol=8) # to store curves
```
> bcoef <- unlist(ranef(fit1))
> indx <- -1 + trdata$trt + 2*trdata$site  #random treatment effect index
> Ainv <- diag(1/rep(unlist(VarCorr(fit1)), c(8,4)))
> for (i in 1:4) {
  tcoef <- bcoef
  for (j in 1:101) {
    tcoef[i+8] <- xx[j]  #reset single coef
    eta <- fixef(fit1)*trdata$trt + tcoef[trdata$site+8] + tcoef[indx]
    tfit <- coxph(Surv(futime, status) ~ offset(eta), data= trdata)
    profile[j,i] <- tfit$loglik - .5*tcoef%*% Ainv %*% tcoef
    profile[j, i+4] <- fit1$loglik[3] - .5*sum(((tcoef-bcoef) %*% fit1$hmat[1:12, 1:12])^2)
  }
}
> matplot(xx, profile-fit1$loglik[3], type='l', lty=c(1,1,1,1,2,2,2,2), col=1:4,
          ylim=c(-40,0), xlab="b", ylab="LPPL - max")

One returned component of coxme is hmat, which contains the generalized Cholesky decomposition \(LDL'\) of \(H\), based on the bdsmatrix library. To take advantage of sparse matrices, the coxme code orders the coefficients as \((b, \beta)\), so we want the upper left portion of \(H\) in our code. A product \(x \%\%\% fit1$hmat\) returns \(y = xLD^{1/2}\), then \(yy' = xHx' = \text{sum}(y^2)\).

### 3.2 Control sampling

To evaluate the integral numerically we use variance reduction methods. Control sampling is based on the simple equation

\[
C = B + (C - B)
\]

In this case \(C\) is the desired integral, the right hand side of equation (1), \(B\) is the Laplace approximation to the integral, and we simulate \(C - B\).

\[
C - B = n(A) \int e^{LPPL(\hat{\beta}, b) - b' A^{-1} b/2} - e^{LPPL(\hat{\beta}, \hat{b}) - (b-\hat{b})' H_{bb}(\hat{b}-\hat{b})/2} db
\]

\[
= n(A) e^k \int e^{LPPL(\hat{\beta}, b) - k} - e^{LPPL(\hat{\beta}, \hat{b}) - (b-\hat{b})' H_{bb}(\hat{b}-\hat{b})/2-k} \frac{g(b)}{g(\hat{b})} db
\]

In equation (2) we expect the integrand to be close to zero for all values of \(b\). Since the variance of our Monte Carlo result is the variance of this integrand divided by the number of simulations, the Monte Carlo result will also be precise. A Monte Carlo evaluation with respect to the vague prior \(db\) is not possible, however, and equation (3) rewrites this so that we sample from a distribution \(g(b)\). The divisor \(\exp(k)\) is chosen to keep the arguments of the two exponentials in bounds and avoid underflow/overflow errors.
The choice of $g$ is important. We want to make sure that $g$ is never tiny when the numerator is near its largest values, as that would generate large values and erase much of the good done by the control function. Both exponentials reach their maximum at $\hat{b}$ so it seems sensible to center $g$ there. The difference in the numerator can be no bigger than the smaller of the two exponentials, so a distribution that falls away a little more slowly than the right hand quadratic term would add the desired margin of safety. A natural choice satisfying these two is a multivariate t-distribution with variance matrix $H^{-1}_{22}$ and a modest degrees of freedom.

Control sampling has been incorporated into coxme and is invoked by the `refine.n` option. The result for our sample data set is shown below.

```r
> fit1b <- coxme(Surv(futime, status) ~ trt + (1 | site/trt),
data=trdata, refine.n=500)
> fit1b$refine
  correction std
     -0.0004463318  0.0024278714
```

This verifies what the figure implied: the Laplace approximation is excellent for this data set. The eventual test for significance of the random effects will be based on a chi-square distribution with 2 degrees of freedom, comparing the IPL for the fitted coxme model to the PL from a fixed effects coxph model with treatment alone. This suggests that any error in the IPL that is less than 0.1 will be of little import.

### 4 Further examples

#### 4.1 EORTC

As a larger example consider the eortc data set. This is a simulated data set, but based on a large breast cancer clinical trial. There are 37 enrollment centers, enrolling from 21 to 247 patients each.

```r
> efit2 <- coxme(Surv(y, uncens) ~ trt + (1|center), eortc,
  refine.n=100)
> efit2$refine
  correction std
     -0.04333163 0.04608217
> efit3 <- coxme(Surv(y, uncens) ~ trt + (1|center/trt), eortc,
  refine.n=100)
> efit3$refine
  correction std
     -0.04206053 0.10112370
```

```r
cox mixed-effects model fit by maximum likelihood
Data: eortc
events, n = 1463, 2323
Iterations= 10 54
```
null integrated fitted

log-likelihood -10638.71 -10517.57 -10464.38

chisq df p aic bic

integrated loglik 242.28 3.00 0 236.28 220.42
penalized loglik 348.67 30.96 0 270.16 61.56

model: surv(y, uncens) ~ trt + (1 | center/trt)

fixed coefficients

trt 0.7420388 2.100213 0.05270483 8.97 0

random effects

group variable std dev variance

center/trt (intercept) 0.20451052 0.04182455

center (intercept) 0.26273062 0.06902738

this behavior has been the norm for the author's experience with coxme. however, note that the total number of events 2323 is much larger than the effective degrees of freedom for the model of 39.3. we will return to this point.

4.2 ridge regression

a classical method in linear models is ridge regression, which solves the penalized regression problem

$$
\min_{\beta} ||y - X\beta||^2 + \lambda \sum_{j=1}^{p} \beta_j^2
$$

this penalizes large values of the coefficients and can stabilize problems with near collinear x matrices. as \( \lambda \) goes to zero we approach the ordinary least squares result, as \( \lambda \) increases coefficients are shrunken towards zero. the intercept term \( \beta_0 \) is normally left out of the penalty.

the penalty can also be viewed as imposing a gaussian prior on the coefficients. similarly, we can use coxme to perform ridge regression cox models. we will use an advanced lung cancer data set as our example, it is part of the survival package.

```r
> lfit1 <- coxph(surv(time, status) ~ age + ph.ecog + wt.loss, lung)
> lfit2 <- coxme(surv(time, status) ~ age + (ph.ecog |1) +
  (wt.loss |1), data=lung, refine=n=100)
```

```r
> lfit2$refine

<table>
<thead>
<tr>
<th>correction</th>
<th>std</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0004929546</td>
<td>0.0002753108</td>
</tr>
</tbody>
</table>
```

again, the laplace transform works very well. by default the random coefficients \( b \) are no included in the printout, but they can be requested with the rcoef option.

```r
> print(lfit2, rcoef=TRUE)
```
Cox mixed-effects model fit by maximum likelihood
Data: lung
events, n = 151, 213 (15 observations deleted due to missingness)
Iterations= 15 63

Log-likelihood -675.0244 -667.9031 -666.0607

Chisq df p AIC BIC
Integrated loglik 14.24 3.00 0.0025928 8.24 -0.81
Penalized loglik 17.93 2.01 0.0001291 13.92 7.86

Model: Surv(time, status) ~ age + (ph.ecog | 1) + (wt.loss | 1)
Fixed and penalized coefficients

age 0.014499269 1.0146049 0.009777229 1.48 0.14
1.ph.ecog 0.397337119 1.4878574 0.116876916
1.wt.loss -0.000542036 0.9994581 0.001868676

Random effects

Group Variable Std Dev Variance
1 (Shrinkage) 4.138505e-01 1.712723e-01
1 (Shrinkage) 1.954502e-03 3.820078e-06

> signif(rbind(coef(lfit1),
c(fixef(lfit2), unlist(ranef(lfit2)))),2)
age ph.ecog wt.loss
[1,] 0.013 0.47 -0.00720
[2,] 0.014 0.40 -0.00054

Again, the Laplace transform works very well. Contrasting the coefficients between the
shrunken and the regular Cox models, the coefficient for weight loss has been reduced over 10
fold while that for ECOG performance score has changed only a little. Weight loss is a weak
predictor in this data set and shrinking it has only a small effect on the fit, whereas performace
score has a much stronger relationship to survival.

4.3 Chronic Granulomotous Disease

Children with chronic granulomotous disease are subject to repeated infections due to an immune
defect. The CGD data set is based on a placeob controlled randomized trial of gamma interferon
for reduction of the infection frequency, during the course of the study enrolled subjects experi-
enced 0–7 infections each. For further discussion of the data see Therneau and Grambsch [1].

This data set is a much stiffer challenge for the Laplace approximation since there are both
a much larger number of random effects (128 subjects) and we do not have “a large number of
events” per random effect. Over half of subjects, and hence their corresponding coefficients \(b_i\),
have none!
\[ \sqrt{\exp(e_2)} > \sqrt{\exp(e_1)} \]

Figure 2: The two terms in the control function Monte Carlo \( \exp(e_1) - \exp(e_2) \).

\begin{verbatim}
> cfit <- coxme(Surv(tstart, tstop, status) ~ treat + age +
>   (1 | id), data=cgd, refine.n=500, refine.detail=TRUE)
> cfit$refine
correction       std
   0.39534744 0.08288391
> 2*(diff(cfit$loglik[1:2]))
Integrated
   34.53227
> temp <- cfit$refine.detail
> e1 <- (temp$loglik - temp$penalty1) - cfit$loglik[2]
> ssqrt <- function(x) sign(x)*sqrt(abs(x)) #signed square root
\end{verbatim}
The Laplace approximation to the IPL is off by 1-2% of the difference between the null and fitted model. In this particular case it is not a severe issue, the test statistic for significance is $> 9$ on 1 df, and even if it were not “significant” a correction for within-subject correlation is called for. Figure 2 shows the two terms of equation (3), plotted on a square root scale to spread the data out. In spite of pushing the approximation to its limit, the quadratic approximation is still working remarkably well. (The refine.detail option was intendend for debugging the code, but the returned information can sometimes be useful for digging deeper.)

A more interesting question is what impact this error might have on the estimate of $\theta$. We can investigate this by looking at a set of fixed variances. The result is shown in figure 3. What we see is that the error increases with the variance of the random effect and the the overall impact is to underestimate the variance: the red (true IPL) maximizes to the right of the maximized Laplace value. The increase in bias is not surprising: as the variance increases the distance from the origin over which we are expecting the approximation to hold also increases. The second is a subject of further investigation. The horizontal line is 3.84/2 units below the Laplace maximum, its intersection with the curve describes a 95% confidence interval for the standard deviation of the random effect. The code to create the figure is shown below.

```
> ss <- seq(.3, 1.3, length=25)
> tmat <- matrix(0, nrow=25, ncol=3)
> for (i in 1:25) {
    tfit <- coxme(Surv(tstart, tstop, status) ~ treat + age + (1|id),
                  cgd, vfixed=ss[i]^2, refine.n=1000)
    tmat[i,] <- c(diff(tfit$loglik[1:2]), tfit$refine)
}
> temp1 <- tmat[,1] + tmat[,2] #corrected IPL
> temp2 <- tmat[,1] + tmat[,2] + cbind(-2*tmat[,3], 2*tmat[,3]) # .955 CI
> matplot(ss, cbind(tmat[,1], temp1), pch='o', col=1:2,
          ylim=range(tmat[,1], temp2), xlab="Std of random effect",
          ylab="IPL - Null")
> segments(ss, temp2[,1], ss, temp2[,2], lty=2, col=2)
> lines(smooth.spline(ss, temp1, df=5), col=2)
> abline(h=diff(cfit$loglik[1:2]) - qchisq(.95, 1)/2, lty=2)
```

### 4.4 Colon cancer data

The colon cancer data set (from the survival package) gives progression and death times of 929 subjects enrolled in a 3 arm clinical trial. A joint analysis of the two outcomes should adjust for fact that subject observations are correlated: in fact they are extremely correlated given the nature of the disease. An estimating equation model is our first choice.

```
> cfit1 <- coxph(Surv(time, status) ~ rx + nodes + extent +
                strata(etype) + cluster(id), colon)
> cfit1
```
Call:
coxph(formula = Surv(time, status) ~ rx + nodes + extent + strata(etype) +
        cluster(id), data = colon)

    coef exp(coef) se(coef) robust se  z   p
rxLev  -0.08499  0.91852  0.07796  0.10625 -0.80 0.42
rxLev+5FU -0.46860  0.62588  0.08520  0.11543 -4.06 4.9e-05
nodes   0.08367  1.08727  0.00634  0.01267 6.61 3.9e-11
extent  0.54750  1.72892  0.07992  0.10557 5.19 2.1e-07

Likelihood ratio test=222.8 on 4 df, p=<2e-16
n= 1822, number of events= 897
(36 observations deleted due to missingness)

The fitted model shows no difference between the levamisole and observation arms, an important
decrease in risk for the combination therapy levamisole + 5FU, and, as expected, large effects
for the number of lymph nodes and the extent of tumor invasion. The reduction in standard
error between the model based and robust variance is almost √2, which is what we would get if
the two outcomes were perfectly redundant. A per subject random effect is not sensible when
there is only 1 event per subject, which is what we effectively have. Nevertheless, we will fit and
examine the result.

> cfit2 <- coxme(Surv(time, status) ~ rx + nodes + extent +
                   strata(etype) + (1|id), colon,
                   refine.n=500)
> cfit2$refine
  correction std
    -1.3848985  0.4820886
> print(cfit2)
Cox mixed-effects model fit by maximum likelihood
Data: colon
events, n = 897, 1822 (36 observations deleted due to missingness)
Iterations= 33 252

       NULL Integrated Fitted
Log-likelihood -5804.469 -5261.436 -4056.941

               Chisq df p   AIC    BIC
Integrated loglik 1086.07 5.00 0 1076.07 1052.07
Penalized loglik 3495.06 725.78 0 2043.49 -1439.60

Model: Surv(time, status) ~ rx + nodes + extent + strata(etype) + (1 | id)
### Fixed coefficients

<table>
<thead>
<tr>
<th>Variable</th>
<th>coef</th>
<th>exp(coef)</th>
<th>se(coef)</th>
<th>z</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>rxLev</td>
<td>0.01022</td>
<td>1.01027</td>
<td>0.25148</td>
<td>0.04</td>
<td>9.7e-01</td>
</tr>
<tr>
<td>rxLev+5FU</td>
<td>-0.7915</td>
<td>0.45315</td>
<td>0.25341</td>
<td>-3.12</td>
<td>1.8e-03</td>
</tr>
<tr>
<td>nodes</td>
<td>0.2404</td>
<td>1.27172</td>
<td>0.02742</td>
<td>8.77</td>
<td>0.0e+00</td>
</tr>
<tr>
<td>extent</td>
<td>1.2016</td>
<td>3.32550</td>
<td>0.22983</td>
<td>5.23</td>
<td>1.7e-07</td>
</tr>
</tbody>
</table>

### Random effects

<table>
<thead>
<tr>
<th>Group Variable</th>
<th>Std Dev</th>
<th>Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>2.756</td>
<td>7.669</td>
</tr>
</tbody>
</table>

> round(quantile(ranef(cfit2)[[1]], 0:8/8), 2)

<table>
<thead>
<tr>
<th></th>
<th>0%</th>
<th>25%</th>
<th>50%</th>
<th>75%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-6.14</td>
<td>-2.34</td>
<td>-2.01</td>
<td>-1.60</td>
<td>-0.73</td>
</tr>
<tr>
<td></td>
<td>0.63</td>
<td>1.75</td>
<td>3.09</td>
<td>8.84</td>
<td></td>
</tr>
</tbody>
</table>

The variance of the random effects is very large at 7.6. Subjects have estimated random effects of \( \exp(-6) = 0.002 \) (nearly immortal) to \( \exp(8) > 6600 \) (dies before getting out of the building), which are biologically implausible. The control based refinement was not able to reliably estimate the bias — for many values of the random number seed it actually returns NA due to computations that go out of range. A mixed effects model has not been successful for this data set.

### 4.5 Genetic studies

Data sets that contain genetic correlations was actually the genesis of the coxme function, therefore the performance of the Laplace in this case is of particular interest to us.

The story here is still being worked out and we expect to have a more detailed description in future versions of this vignette. In broad strokes, when the correlation is based on a kinship matrix the Laplace appears to work adequately when: family sizes are modest to large and the standard deviation of the random effect is no greater than .8-.9. Even though there is a random effect per subject in such models, the correlation structure is such that each random effect is “linked” to a sufficiently number of events. More work with a range of data sets needs to be undertaken, however.

### 5 Conclusions

On many data sets the Laplace works very well, on others it is adequate, and there are a few where fails. An example of the last is the colon cancer data set. However, this is a data set for which I have grave doubts about the applicability of a a mixed effects model at all, a reservation that extends to any data set where the effective degrees of freedom approaches the total number of events.

For the case of generalized linear models, Shun and McCullaugh [2] suggest that the ordinary Laplace approximation will be sufficient when the degrees of freedom for the random effect is \( o(\sqrt{n}) \). For survival models experience with other cases such as AIC suggests that the appropriate \( n \) for such calculations is the number of deaths. Hall et al [3] point out one of the reasons for difficulty when there are a large number of random effects. The average distance from the center for a multivariate Gaussian in \( d \) dimensions is \( \sqrt{d} \). For large \( d \) the law of large numbers...
guarantees that essentially all the mass in the distribution is in a narrow annulus $\sqrt{d}$ units from the center. Consequently, this is the distance at which we are demanding accuracy from the quadratic approximation when we use the Laplace method. For the colon data we have 911 random effects and a variance of 7.6. The distance from the origin is over 83 units which is too much to ask of a second order Taylor series.

For our examples we had the following for number of events and effective degrees of freedom:

<table>
<thead>
<tr>
<th></th>
<th>Events</th>
<th>EDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation</td>
<td>480</td>
<td>4.1</td>
</tr>
<tr>
<td>cortc</td>
<td>2323</td>
<td>39.3</td>
</tr>
<tr>
<td>Ridge regression</td>
<td>151</td>
<td>2</td>
</tr>
<tr>
<td>CGD</td>
<td>76</td>
<td>29.3</td>
</tr>
<tr>
<td>colon cancer</td>
<td>897</td>
<td>725.8</td>
</tr>
</tbody>
</table>

The success with the CGD data suggest that for a mixed effects Cox model at least, the Shun and McCullaugh bounds may be overly conservative. Checking the reliability of the Laplace through use of the refine.n option is encouraged.

References


Figure 3: The Laplace approximation to the IPL for the CGD data is in black for a range of trial variances, along with a Monte Carlo correction to this in red. The vertical red lines are ±2 times the estimated Monte Carlo standard standard error.