Coxme and the Laplace Approximation

Terry Therneau
Mayo Clinic

December 28, 2011

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 λ_0 is an unspecified baseline hazard function, X and Z are the design matrices for the fixed and random effects, respectively, β 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 Σ, which in turn depends a vector of parameters θ.

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

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

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 eplacement of the log penalized partial likelihood $LPPL$ with a second order Taylor series about its value at the maximum of the function

$$PL(\beta, b) = e^{\log(PL(\beta,b))} \equiv e^{LPL(\beta,b)}$$
$$LPPL(\beta, b, \theta) = LPL(\beta, b) - (1/2)b'A^{-1}(\theta)b$$
$$\approx LPPL(\hat{\beta}(\theta), \hat{b}(\theta)) - (1/2)(\beta - \hat{\beta}(\theta), b - \hat{b}(\theta))'H(\beta - \hat{\beta}(\theta), b - \hat{b}(\theta))$$

where the Hessian H is -1 times the matrix of second derivatives of the LPPL, evaluated at $(\hat{\beta}(\theta), \hat{b}(\theta))$. When θ and hence $A(\theta)$ are fixed, the relevant values of β 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 $(0, b - \hat b)' H (0, b - \hat b) = (b - \hat b)' H_{bb} (b - \hat b)$, 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 `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 `coxme` is an outer and an inner loop. The outer loop searches over the parameters θ of the variance matrix for a maximum of the IPL. For each trial value of θ 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 $|H|$ of the Hessian 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.

```
> 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)
        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(var(c(.5, 1.5, 2, 1)))` = .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
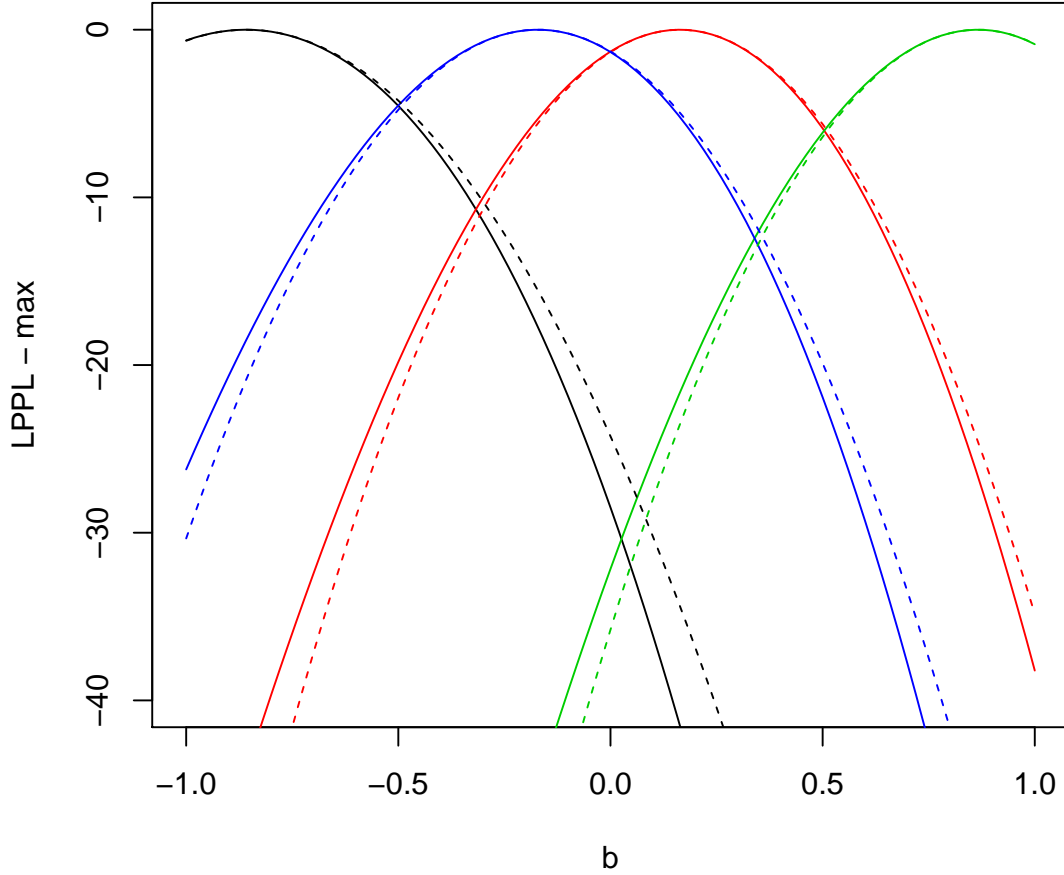
Figure 1: The solid lines are profiles of $PPL(\hat{\beta}, b)$, dashed lines are the Taylor series approximation.

"slices" through the 12 dimensional LPPL as a function of b. The approximation is not perfect — 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(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 $\sum_1^8 b_i = 0$ and $\sum_9^{12} b_i = 0$.

4

```
> 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, β), so we want the upper left portion of H in our code. A product `x %*% fit$hmat` returns $y = xLD^{1/2}$, then $yy' = xHx' = $ `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 $C - B$.

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

$$= n(A)e^{k} \int \frac{e^{LPPL(\hat{\beta}, b) - k} - e^{LPPL(\hat{\beta}, \hat{b}) - (b - \hat{b})' H_{bb}(b - \hat{b})/2 - k}}{g(b)} g(b) db \qquad (3)$$

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 it's 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_{22}^{-1} 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.

```
> 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. A test for significance of the random effects will be based on the chisquare 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 examles

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.

```
> 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
> efit3
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 39.26 0 270.16  62.56

Model:  Surv(y, uncens) ~ trt + (1 | center/trt)
Fixed coefficients
        coef exp(coef)   se(coef)    z p
trt 0.7420388  2.100213 0.08270483 8.97 0

Random effects
 Group      Variable   Std Dev    Variance
 center/trt (Intercept) 0.20451053 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 λ goes to zero we approach the ordinary least squares result, as λ increases coefficients are shrunken towards zero. The intercept term β_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 surival package.

```
> 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)
> lfit2$refine
   correction          std
0.0004929546 0.0002753107
```

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.

```
> 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
                     NULL Integrated    Fitted
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
                   coef exp(coef)    se(coef)    z    p
age        0.0144992691 1.0146049 0.009777229 1.48 0.14
1.ph.ecog  0.3973371185 1.4878574 0.116876916
1.wt.loss -0.0005420359 0.9994581 0.001868676

Random effects
 Group Variable     Std Dev        Variance
 1      (Shrinkage) 4.138505e-01 1.712723e-01
 1      (Shrinkage) 1.954502e-03 3.820077e-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 coeffienct for weight loss has been reduced over 10 fold while that for ECOG performace 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 Granulotomous Disease

Children with chronic granulotomous 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 experienced 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!
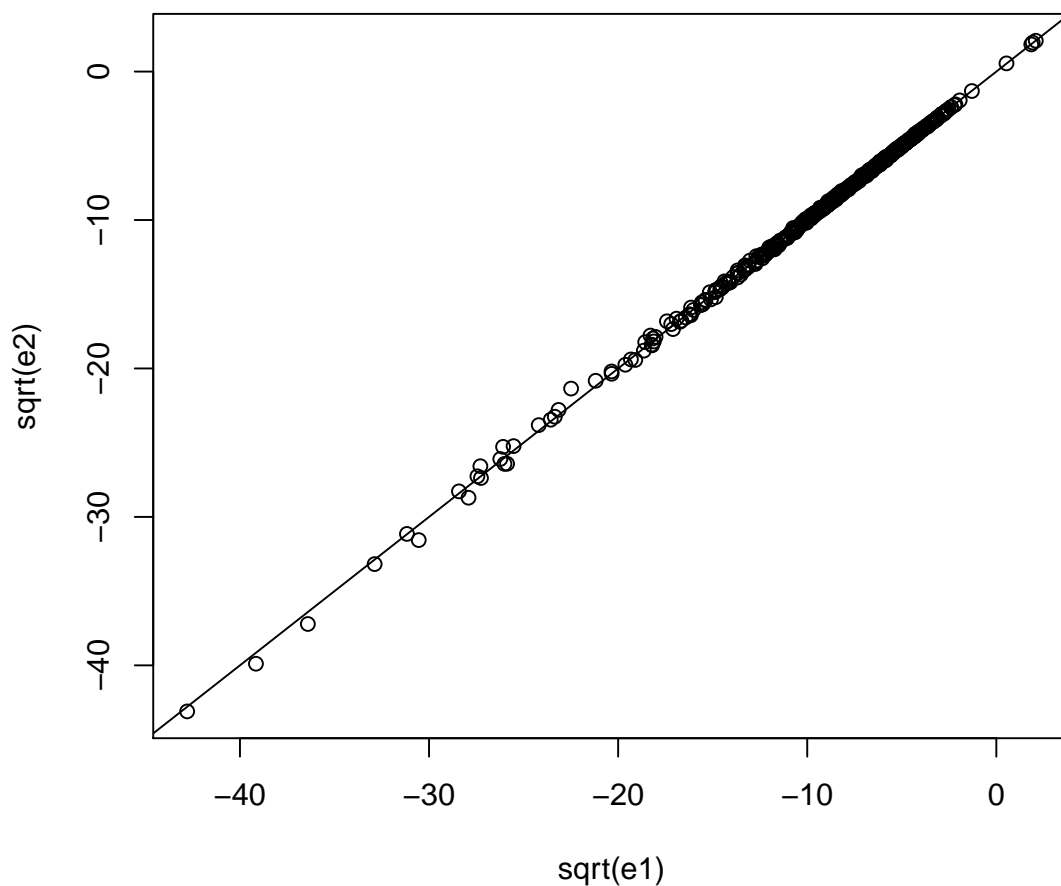
Figure 2: The two terms in the control function Monte Carlo $\exp(e1) - \exp(e2)$.

```
> 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]
> e2 <- (cfit$loglik[3] - temp$penalty2) - cfit$loglik[2]
> ssqrt <- function(x) sign(x)*sqrt(abs(x))  #signed square root
```

```
> plot(ssqrt(e1), ssqrt(e2), xlab="sqrt(e1)", ylab="sqrt(e2)")
> abline(0,1)
```

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 "singnificant" 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 it's limit, the 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 θ. 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
```

10

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


             coef exp(coef) se(coef) robust se     z
rxLev     -0.0850     0.919  0.07796    0.1063 -0.80
rxLev+5FU -0.4686     0.626  0.08520    0.1154 -4.06
nodes      0.0837     1.087  0.00634    0.0127  6.61
extent     0.5475     1.729  0.07992    0.1056  5.19
                 p
rxLev     4.2e-01
rxLev+5FU 4.9e-05
nodes     3.9e-11
extent    2.1e-07

Likelihood ratio test=223  on 4 df, p=0  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 $\sqrt{2}$, 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= 33 251
                    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
                 coef exp(coef)  se(coef)      z        p
rxLev       0.01022041 1.0102728 0.2514787  0.04 9.7e-01
rxLev+5FU  -0.79153227 0.4531499 0.2534126 -3.12 1.8e-03
nodes       0.24037038 1.2717201 0.0274218  8.77 0.0e+00
extent      1.20162038 3.3255011 0.2298298  5.23 1.7e-07

Random effects
 Group Variable  Std Dev  Variance
 id    Intercept 2.755646 7.593586
> round(quantile(ranef(cfit2)[[1]], 0:8/8), 2)
   0% 12.5%   25% 37.5%   50% 62.5%   75% 87.5%  100%
-6.14 -2.34 -2.01 -1.60 -0.73  0.63  1.75  3.09  8.84
```

The variance of the random effects is very large at 7.6. Subjects have estimated random effects of $\exp(-6.1) = .002$ (nearly immortal) to $\exp(8.8) > 6600$ (dies before getting out of the building) which are biologically implausible. The control based refinement was not able to reliable 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 performace 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 large modest to large, 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 Lapalce 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 modeled with a random effect per subject but having only one observation per subject.

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 $o(\sqrt[3]{n})$. 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

guarrantees 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:

	Events	EDF
Simulation	480	4.1
eortc	2323	39.3
Ridge regression	151	2
CGD	76	29.3
colon cancer	897	725.8

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 reliabilty of the Laplace through use of the refine.n option is encouraged.

References

[1] T. Therneau and P. Grambsch, Modeling Survival Data: Extending the Cox Model, Springer-Verlag, 2000.

[2] Z. Shun and P. McCullaugh, The Laplace approximation for high dimensional integrals, JRSSB 57:749–760, 1995.

[3] P. Hall and J.S. Marron and A Neeman, Geometric representation of high dimension, low sample size data. JRSSB: 67, 427–444, 2005.
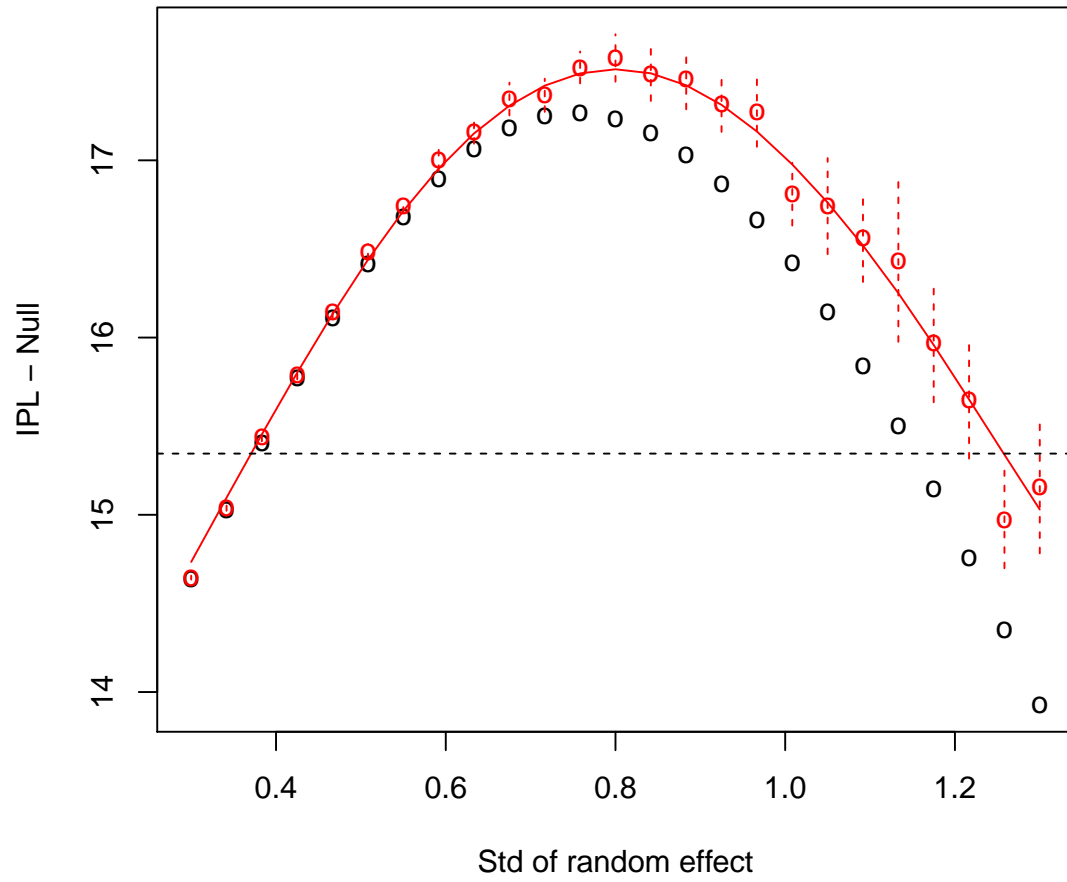
Figure 3: The Laplace approximation to the IPL for the CGD data is in black for a range of trial variances,, along with a Monte Carlo correction to this in red. The vertical red lines are ± 2 times the estimated Monte Carlostandard standard error.