STA465 Final Exam

Tianyi Zhang

29/04/2022

Fitting Models

```r
#### Fit models ####
# Using Default Priors
##### Complete pooling and altitude covariate (no random effect) #####
formula1 <- prev ~ 1 + alt
res1 <- inla(formula1, data = gambia.sf,
            control.predictor = list(compute = TRUE),
            control.compute = list(cpo=TRUE))

##### Hierarchical random effect (iid) - (intercept only) #####
# Include an unstructured component with index variable
gambia.sf$idu <- 1:nrow(gambia.sf)
gambia.sf$idv <- 1:nrow(gambia.sf)

formula2 <- prev ~ 1 + f(idv, model = "iid")
res2 <- inla(formula2, data = gambia.sf,
            control.predictor = list(compute = TRUE),
            control.compute = list(cpo=TRUE))

##### Hierarchical random effect (iid) + altitude covariate #####
formula3 <- prev ~ 1 + alt + f(idu, model = "iid") +
  f(idv, alt, model = "iid")
res3 <- inla(formula3, data = gambia.sf,
            control.predictor = list(compute = TRUE),
            control.compute = list(cpo=TRUE))

##### Spatial + iid random effect ; Default Priors #####
## Mesh Construction
# Code modified from MalariaPrevalence.Rmd in Lab 7
coo <- cbind(d$long, d$lat)
mesh <- inla.mesh.2d(
  loc = coo, max.edge = c(0.1, 5),
  cutoff = 0.01
)

## Construct "A" projection matrices
# SPDE with default priors
spde <- inla.spde2.matern(mesh = mesh, alpha = 2, constr = TRUE)
```

```
indexs <- inla.spde.make.index("s", spde$n.spde)
A <- inla.spde.make.A(mesh = mesh, loc = coo)

## Prediction data
dp <- rasterToPoints(r)
ra <- aggregate(r, fact = 5, fun = mean)
dp <- rasterToPoints(ra)
coop <- dp[, c("x", "y")]
Ap <- inla.spde.make.A(mesh = mesh, loc = coop)

## stack for estimation stk.e
stk.e <- inla.stack(
  tag = "est",
  data = list(y = d$positive, numtrials = d$total),
  A = list(1, A),
  effects = list(data.frame(b0 = 1, altitude = d$alt), s = indexs)
)

# stack for prediction stk.p
stk.p <- inla.stack(
  tag = "pred",
  data = list(y = NA, numtrials = NA),
  A = list(1, Ap),
  effects = list(data.frame(b0 = 1, altitude = dp[, 3]),
    s = indexs
  )
)

# stk.full has stk.e and stk.p
stk.full <- inla.stack(stk.e, stk.p)

###### Model formula and inla call ######
formula4.1 <- y ~ 0 + b0 + f(s, model = spde)

res4.1 <- inla(formula4.1,
  data = inla.stack.data(stk.full),
  control.predictor = list(
    compute = TRUE,
    A = inla.stack.A(stk.full)
  ),
  control.compute = list(cpo=TRUE)
)

##### Spatial + iid random effect ; PC Priors #####
## Mesh Construction
# Code modified from MalariaPrevalence.Rmd in Lab 7
coo <- cbind(d$long, d$lat)
mesh <- inla.mesh.2d(
  loc = coo, max.edge = c(0.1, 5),
  cutoff = 0.01
)

## Construct "A" projection matrices
```

```r
# SPDE2 Matérn with penalized complexity priors
spde.pc <- inla.spde2.pcmatern(mesh = mesh, alpha = 2, constr = TRUE,
                               prior.range = c(10, 0.9), prior.sigma = c(1, 0.01))

indexs <- inla.spde.make.index("s", spde.pc$n.spde)
A <- inla.spde.make.A(mesh = mesh, loc = coo)

## Prediction data
dp <- rasterToPoints(r)
ra <- aggregate(r, fact = 5, fun = mean)
dp <- rasterToPoints(ra)
coop <- dp[, c("x", "y")]
Ap <- inla.spde.make.A(mesh = mesh, loc = coop)

## stack for estimation stk.e
stk.e <- inla.stack(
  tag = "est",
  data = list(y = d$positive, numtrials = d$total),
  A = list(1, A),
  effects = list(data.frame(b0 = 1, altitude = d$alt), s = indexs)
)

# stack for prediction stk.p
stk.p <- inla.stack(
  tag = "pred",
  data = list(y = NA, numtrials = NA),
  A = list(1, Ap),
  effects = list(data.frame(b0 = 1, altitude = dp[, 3]),
    s = indexs
  )
)

# stk.full has stk.e and stk.p
stk.full <- inla.stack(stk.e, stk.p)

###### Model formula and inla call ######
formula4.2 <- y ~ 0 + b0 + f(s, model = spde.pc)

res4.2 <- inla(formula4.2,
  data = inla.stack.data(stk.full),
  control.predictor = list(
    compute = TRUE,
    A = inla.stack.A(stk.full)
  ),
  control.compute = list(cpo=TRUE)
)


##### Spatial + iid random effect + altitude covariate ; Default Priors #####
## Mesh Construction
coo <- cbind(d$long, d$lat)
mesh <- inla.mesh.2d(
  loc = coo, max.edge = c(0.1, 5),
```

```r
  cutoff = 0.01
)

## Construct "A" projection matrices
# SPDE with default priors
spde <- inla.spde2.matern(mesh = mesh, alpha = 2, constr = TRUE)

indexs <- inla.spde.make.index("s", spde.pc$n.spde)
A <- inla.spde.make.A(mesh = mesh, loc = coo)

## Prediction data
dp <- rasterToPoints(r)
ra <- aggregate(r, fact = 5, fun = mean)
dp <- rasterToPoints(ra)
coop <- dp[, c("x", "y")]
Ap <- inla.spde.make.A(mesh = mesh, loc = coop)

## stack for estimation stk.e
stk.e <- inla.stack(
  tag = "est",
  data = list(y = d$positive, numtrials = d$total),
  A = list(1, A),
  effects = list(data.frame(b0 = 1, altitude = d$alt), s = indexs)
)

# stack for prediction stk.p
stk.p <- inla.stack(
  tag = "pred",
  data = list(y = NA, numtrials = NA),
  A = list(1, Ap),
  effects = list(data.frame(b0 = 1, altitude = dp[, 3]),
    s = indexs
  )
)

# stk.full has stk.e and stk.p
stk.full <- inla.stack(stk.e, stk.p)

###### Model formula and inla call ######
formula5.1 <- y ~ 0 + b0 + altitude + f(s, model = spde)

res5.1 <- inla(formula5.1,
  family = "binomial", Ntrials = numtrials,
  control.family = list(link = "logit"),
  data = inla.stack.data(stk.full),
  control.predictor = list(
    compute = TRUE, link = 1,
    A = inla.stack.A(stk.full)
  ),
  control.compute = list(cpo=TRUE)
)

##### Spatial + iid random effect + altitude covariate; PC Priors #####
```

```r
## Mesh Construction
coo <- cbind(d$long, d$lat)
mesh <- inla.mesh.2d(
  loc = coo, max.edge = c(0.1, 5),
  cutoff = 0.01
)

## Construct "A" projection matrices
# SPDE2 Matérn with penalized complexity priors
spde.pc <- inla.spde2.pcmatern(mesh = mesh, alpha = 2, constr = TRUE,
                            prior.range = c(10, 0.9), prior.sigma = c(1, 0.01))


indexs <- inla.spde.make.index("s", spde.pc$n.spde)
A <- inla.spde.make.A(mesh = mesh, loc = coo)

## Prediction data
dp <- rasterToPoints(r)
ra <- aggregate(r, fact = 5, fun = mean)
dp <- rasterToPoints(ra)
coop <- dp[, c("x", "y")]
Ap <- inla.spde.make.A(mesh = mesh, loc = coop)

## stack for estimation stk.e
stk.e <- inla.stack(
  tag = "est",
  data = list(y = d$positive, numtrials = d$total),
  A = list(1, A),
  effects = list(data.frame(b0 = 1, altitude = d$alt), s = indexs)
)

# stack for prediction stk.p
stk.p <- inla.stack(
  tag = "pred",
  data = list(y = NA, numtrials = NA),
  A = list(1, Ap),
  effects = list(data.frame(b0 = 1, altitude = dp[, 3]),
    s = indexs
  )
)

# stk.full has stk.e and stk.p
stk.full <- inla.stack(stk.e, stk.p)

###### Model formula and inla call ######
formula5.2 <- y ~ 0 + b0 + altitude + f(s, model = spde.pc)

res5.2 <- inla(formula5.2,
  family = "binomial", Ntrials = numtrials,
  control.family = list(link = "logit"),
  data = inla.stack.data(stk.full),
  control.predictor = list(
    compute = TRUE, link = 1,
```

```
    A = inla.stack.A(stk.full)
  ),
  control.compute = list(cpo=TRUE)
)
```

CPO + PIT Values

```
#### CPO + PIT values ####
cpo1 <- data.frame(CPO = res1$cpo$cpo,
                       PIT = res1$cpo$pit)
cpo2 <- data.frame(CPO = res2$cpo$cpo,
                       PIT = res2$cpo$pit)
cpo3 <- data.frame(CPO = res3$cpo$cpo,
                       PIT = res3$cpo$pit)
cpo4.1 <- data.frame(CPO = res4.1$cpo$cpo,
                       PIT = res4.1$cpo$pit)
cpo4.2 <- data.frame(CPO = res4.2$cpo$cpo,
                       PIT = res4.2$cpo$pit)
cpo5.1 <- data.frame(CPO = res5.1$cpo$cpo,
                       PIT = res5.1$cpo$pit)
cpo5.2 <- data.frame(CPO = res5.2$cpo$cpo,
                       PIT = res5.2$cpo$pit)
cpo.total <- data.frame(cpo1,cpo2,cpo3,
                        cpo4.1[1:65,],cpo4.2[1:65,],cpo5.1[1:65,],cpo5.2[1:65,])

cpo.total |>
  kable(
    caption = "CPO and PIT values for each Model",
    row.names = FALSE,
    col.names = c("CPO","PIT","CPO","PIT","CPO",
                  "PIT","CPO","PIT",
                  "CPO","PIT","CPO",
                  "PIT","CPO",
                  "PIT"),
    digits = 4,
    booktabs = TRUE,
    align = 'l'
  ) |>
  add_header_above(c("Complete Pool" = 2, "Hierarchical" = 2,
                     "Hierchical+Alt" = 2, "Spatial+Default Priors" = 2,
                     "Spatial+PC Priors" = 2, "Spatial+Alt+Default Priors" = 2,
                     "Spatial+Alt+PC Priors" = 2))
```

Reference

- Asidianya, N, (2022, March 16) Lab Lecture #7. Quercus. https://q.utoronto.ca/
- Barajas, V. Leos, (2022, April 1) Lecture11 [PDF]. Quercus. https://q.utoronto.ca/
- Gómez-Rubio, V. (2021, August 29). Bayesian inference with INLA. Chapter 4 Multilevel Models. Retrieved April 30, 2022, from https://becarioprecario.bitbucket.io/inla-gitbook/ch-multilevel.html

Table 1: CPO and PIT values for each Model

Complete Pool		Hierarchical		Hierchical+Alt		Spatial+Default Priors		Spatial+PC Priors		Spatial+Alt+Default	
CPO	PIT	CPO	PIT	CPO	PIT	CPO	PIT	CPO	PIT	CPO	PIT
1.2933	0.7886	5.4975	0.6065	1.3008	0.8228	0.0428	0.8434	0.0354	0.8841	0.0482	0.7057
1.5363	0.2908	5.5551	0.4259	1.3458	0.3081	0.0223	0.9407	0.0192	0.9484	0.0357	0.2379
1.7790	0.4639	5.4695	0.5241	1.4629	0.4567	0.0432	0.1466	0.0548	0.2248	0.1209	0.7572
1.7689	0.4309	5.6477	0.4538	1.8022	0.4293	0.0700	0.3647	0.0684	0.3535	0.0697	0.7081
1.7799	0.4511	5.4867	0.4985	1.5712	0.4470	0.0747	0.3979	0.0732	0.4522	0.0842	0.3104
1.7730	0.5594	5.4863	0.5026	1.8951	0.5675	0.0530	0.1936	0.0576	0.2422	0.1154	0.4455
0.6681	0.9186	4.7778	0.7012	0.5000	0.9525	0.0013	0.9978	0.0017	0.9969	0.0003	0.9997
0.7181	0.0890	4.6440	0.2845	0.7519	0.1150	0.0552	0.2057	0.0583	0.2468	0.0000	0.0000
0.4296	0.0469	4.4515	0.2673	0.5641	0.0978	0.0593	0.2409	0.0560	0.2305	0.0000	0.0000
1.3921	0.2445	5.5763	0.4313	1.1409	0.2850	0.0743	0.3991	0.0698	0.3693	0.0804	0.7669
1.5230	0.2922	5.2442	0.3452	1.7539	0.2669	0.0622	0.2545	0.0571	0.2360	0.0869	0.3627
1.7876	0.4610	5.4846	0.4895	1.6658	0.4571	0.0714	0.6561	0.0695	0.6373	0.0396	0.9014
1.7149	0.6190	5.5068	0.5469	1.7458	0.6210	0.0449	0.8509	0.0508	0.8044	0.0377	0.9022
1.6490	0.3455	5.5607	0.3909	1.8546	0.3343	0.0433	0.1413	0.0439	0.1545	0.1323	0.4036
0.9715	0.1362	4.3262	0.2571	0.9382	0.1062	0.0543	0.2008	0.0490	0.1827	0.0000	0.0000
1.1008	0.1628	4.7743	0.2963	1.1103	0.1450	0.0366	0.1108	0.0362	0.1169	0.0540	0.1396
1.7138	0.3916	5.5553	0.4008	1.9949	0.3862	0.0750	0.4014	0.0701	0.3732	0.0134	0.0337
0.7100	0.9122	3.8436	0.7783	0.7970	0.8714	0.0346	0.8967	0.0413	0.8569	0.0102	0.9895
1.4048	0.2427	5.3823	0.3626	1.4335	0.2356	0.0574	0.2228	0.0543	0.2173	0.0764	0.2340
1.5618	0.6864	5.4812	0.5130	1.6820	0.7547	0.0646	0.7182	0.0675	0.6615	0.0722	0.5862
1.4630	0.7377	5.4173	0.5998	1.5019	0.7509	0.0357	0.8898	0.0440	0.8432	0.0394	0.8936
1.6347	0.3446	5.5659	0.4854	1.2638	0.3644	0.0752	0.5853	0.0734	0.5331	0.0791	0.3283
1.4777	0.2753	5.6477	0.4538	1.1683	0.3121	0.0748	0.5955	0.0734	0.5357	0.0819	0.6563
0.6228	0.0740	3.6907	0.2061	0.5416	0.0536	0.0191	0.0488	0.0177	0.0461	0.0000	0.0000
1.0457	0.8489	4.5897	0.7182	1.0491	0.7991	0.0503	0.8147	0.0577	0.7531	0.0539	0.8784
1.3481	0.2253	5.3336	0.3563	1.3586	0.2207	0.0748	0.5378	0.0671	0.3494	0.0860	0.4346
1.7692	0.5603	5.4861	0.4936	1.9465	0.5736	0.0355	0.8874	0.0582	0.7356	0.0259	0.9323
0.8143	0.1062	3.9976	0.2303	0.7403	0.0775	0.0557	0.2192	0.0363	0.1195	0.0000	0.0000
0.3414	0.0348	3.4724	0.1889	0.3723	0.0411	0.0384	0.1205	0.0217	0.0597	0.0000	0.0000
0.2552	0.0247	3.4724	0.1889	0.3475	0.0434	0.0384	0.1203	0.0216	0.0597	0.2412	0.2412
0.9909	0.1386	4.8891	0.3072	0.9842	0.1441	0.0757	0.5485	0.0677	0.3571	0.0252	0.5078
1.2638	0.2048	4.8719	0.3055	1.3343	0.1742	0.0711	0.3457	0.0528	0.2112	0.1016	0.4149
1.3126	0.2168	5.0720	0.3260	1.3832	0.1950	0.0771	0.4875	0.0663	0.3320	0.1080	0.7362
0.8938	0.1201	4.3546	0.2593	0.8685	0.1025	0.0608	0.2438	0.0428	0.1501	0.0190	0.0423
1.7936	0.4768	5.5604	0.4793	1.7965	0.4751	0.0441	0.8527	0.0583	0.7438	0.0392	0.8923
1.0361	0.1497	4.4725	0.2691	1.0225	0.1197	0.0519	0.1855	0.0388	0.1291	0.0484	0.0877
1.4354	0.2531	5.5553	0.4008	1.3109	0.2696	0.0765	0.5491	0.0721	0.4307	0.1000	0.5787
1.5556	0.3024	5.3823	0.3626	1.7613	0.2828	0.0740	0.3829	0.0627	0.2871	0.1250	0.5578
1.7358	0.4210	5.5630	0.4022	2.0922	0.4225	0.0575	0.7729	0.0665	0.6619	0.0554	0.6303
0.8658	0.1201	5.5607	0.3909	0.8218	0.2030	0.0735	0.4094	0.0660	0.3246	0.0738	0.4338
1.7281	0.3933	5.6813	0.4266	1.8485	0.3897	0.0692	0.3438	0.0699	0.3761	0.0499	0.3548
1.0633	0.8453	5.2798	0.6459	1.0193	0.8742	0.0620	0.7435	0.0554	0.7760	0.0588	0.7149
1.3181	0.7825	5.1699	0.6609	1.2523	0.7436	0.0668	0.7011	0.0574	0.7612	0.0516	0.8079
1.7697	0.4434	5.5850	0.4336	1.9890	0.4450	0.0414	0.1314	0.0496	0.1891	0.1257	0.5487
1.7888	0.4861	5.5220	0.4564	1.9804	0.4916	0.0281	0.9230	0.0247	0.9305	0.0718	0.6044
1.2428	0.1962	5.2904	0.3507	1.2195	0.2010	0.0216	0.0551	0.0307	0.0949	0.0538	0.1513
1.2269	0.8076	5.3560	0.6363	1.2337	0.8170	0.0772	0.5555	0.0716	0.6015	0.0683	0.3629
0.2021	0.9813	2.5328	0.8668	0.1906	0.9830	0.0391	0.1210	0.0464	0.1705	0.2278	0.7465
0.0602	0.9953	1.3998	0.9328	0.0685	0.9945	0.0773	0.4672	0.0737	0.5190	0.0554	0.9828
1.7428	0.5959	5.4584	0.5312	1.8025	0.6004	0.0661	0.2861	0.0715	0.4097	0.0640	0.3842
0.9379	0.8720	4.7606	0.7028	0.9558	0.8619	0.0553	0.7977	0.0462	0.8342	0.0699	0.6196

- Moraga, P. (2020). Geospatial Health Data: Modeling and visualization with R-INLA and shiny. Chapter 9 Spatial modeling of geostatistical data. Malaria in The Gambia. Retrieved April 30, 2022, from https://www.paulamoraga.com/book-geospatial/sec-geostatisticaldataexamplespatial.html