**remote sensing LMM 3Dec21**

I used the LMM simulation model I have to look at the model structure in the AIST2 proposal. The model is

*yi*(*t*) = (*xi b* + f*i*)*t* + e*i*(*t*)

e*i*(*t*) = be*i*(*t*–1) + d*i*(*t*)

d*i*(*t*) ~ N(0, s2**S**)

f*i* ~ N(0, s2f**S**f) (Eq. 1)

where *yi*(*t*) is the value of interest (e.g., NDVI) in pixel (location) *i* in year *t* (*t* = 1, 2, ..., *T*). The time trend that depends on a temporally invariant predictor variable *xi* is *xibt*. There is also a random term that affects the time trend, f*it*, where f*i* is a temporally invariant Gaussian random variable. The spatiotemporal random error term, e*i*(*t*), is governed by a Gaussian autoregressive process generated from the normal random variable d*i*(*t*) that has mean zero and variance s2, with values independent through time so that E{d*i*(*t*) d*i*(*s*)} = 0 for *s* ≠ *t*. The dependence of e*i*(*t*) on b*i* e*i*(*t*–1) generates temporal autocorrelation. Spatiotemporal autocorrelation is generated by assuming E{d*i*(*t*) d*j*(*t*)}= exp(-*dij*/range) where *dij* is the distance between pixels *i* and *j*, where distances are standardized by the maximum (corner to corner) size of the map.

I have code that fits this model with ML. I used it to fit four model variants:

*yi*(*t*) = *a* + (*xi bi*)*t* + e*i*(*t*)

e*i*(*t*) = be*i*(*t*–1) + d*i*(*t*)

d*i*(*t*) ~ N(0, s2**S**) (mod 1)

Model 1 contains only spatiotemporal variation.

*yi*(*t*) = *a* + f0*i* + (*xi b* + f1*i*)*t* + e*i*(*t*)

e*i*(*t*) = be*i*(*t*–1) + d*i*(*t*)

d*i*(*t*) ~ N(0, s2**S**)

f0*i* ~ N(0, s2f0'**I**)

f1*i* ~ N(0, s2f1'**I**) (mod 2)

Model 2 contains a time-invariant random effect for the intercept and the time trend, but there is no spatial autocorrelation in these random effects.

*yi*(*t*) = *a* + f0*i* + (*xi b* + f1*i*)*t* + e*i*(*t*)

e*i*(*t*) = be*i*(*t*–1) + d*i*(*t*)

d*i*(*t*) ~ N(0, s2**S**)

f0*i* ~ N(0, s2f0**S**f0 + s2f0'**I**)

f1*i* ~ N(0, s2f1**S**f1 + s2f1'**I**) (mod 3)

Model 3 contains a time-invariant random effect for the intercept and the time trend, including spatial autocorrelation in these random effects. Model 3 is the same as the simulation model, but with fixed and random intercept terms, and not only spatially autocorrelated variation in f0*i* and f1*i*, but also non-spatially autocorrelated variation.

*yi*(*t*) = *a* + (*xi b* + f*i*)*t* + e*i*(*t*)

e*i*(*t*) = be*i*(*t*–1) + d*i*(*t*)

d*i*(*t*) ~ N(0, s2**S**)

f*i* ~ N(0, s2f**S**f) (mod 4)

This is exactly the simulation model, except with an intercept. Several terms from model 3 are set to zero. Ordinarily, I wouldn't fit this without at least a term f0*i* as in model 3 because it makes too many assumptions about the structure of the data, but I wanted to have a model close the simulation model.

All models are fit with ML, and approximate p-values are given by a Wald approximation, conditional of the estimates of s2 and s2f. The simulations are for 30 years on a 6x6 map (36 pixels) because the code is slow. The Wald approximations are expected to give p-values that are too low especially for model 3, because the model is over-parameterized. I ran 300 simulations, and I'll just present the rejection rates for an alpha significance level of 0.05:

mod 1 mod 2 mod 3 mod 4

0.49 0.42 0.18 0.066

All models had inflated type I error rates, but models 3 and 4 were much better. The difference between models 3 and 4 I think is just due to model 3 being overparameterized for the small simulations.

Model 3 is also capable of detecting the existence of fixed spatial variation, s2f0 and s2f1. The logLik of model 3 was on average 4 greater than for model 2.

**Summary**

These simulations highlight two issues. First, statistically separating spatial (f*i*) from spatiotemporal (e*i*(*t*)) variation is possible. This is important, because spatial variation means that residual patterns in the time trends are predictive of future time trends; time trends stay in the same location. Second, the spatiotemporal variable e*i*(*t*) cannot fully account for spatial variation generated by f*i*, and this inflates type I errors.

**LMM vs. PARTS 23Oct21**

I did another simulation to check the power of PARTS against a "gold standard". I did this in 29 January 2021 with a different model structure. Here I use a simpler model. As a gold standard, I'm using a LMM that Mike Bosch is also using on aphid data in which the residuals are ARMA(p,q); it is modified from the function pglmm in the phyr package in R that is designed for phylogenetic analyses, so we've called it pglmm\_ARMA(). The residuals also have spatial autocorrelation in the same form as the simulation model in the RSE manuscript. Specifically, the model is

*xi*(*t*) = a*i* + z*t* + e*i*(*t*)

e*i*(*t*) = b*i*e*i*(*t*–1) + d*i*(*t*) (Eq. 1)

where *xi*(*t*) is the value of interest (e.g., NDVI) in pixel (location) *i* in year *t*, a*i* is the intercept, and z*i* is a coefficient that measures the effect of time *t* on *xi*(*t*), where *t* = 1, 2, ..., *T*. [Note that pglmm\_ARMA can also fit the model in which the time trend is (z + *ci*)*t* where *ci* is a random effect to give the case of fixed spatial variation in the time trend. This is discussed in the note from 29Jan21. In contrast, here I assume z is the same for every location.] Environmental variation that affects changes in *xi*(*t*) from one year to the next is given by the random variable e*i*(*t*). To account for possible temporal autocorrelation, e*i*(*t*) is governed by a normal (Gaussian) autoregressive process generated from the normal random variable d*i*(*t*) that has mean zero and variance s2, with values independent through time so that E{d*i*(*t*) d*i*(*s*)} = 0 for *s* ≠ *t*. The dependence of e*i*(*t*) on b*i* e*i*(*t*–1) generates temporal autocorrelation. Spatiotemporal autocorrelation is generated by assuming E{d*i*(*t*) d*j*(*t*)}= exp(-*dij*/range) where *dij* is the distance between pixels *i* and *j*, where distances are standardized by the maximum (corner to corner) size of the map.

Due to computational, I used a 8x8 map (64 pixels). This still took 24 hours, since the code for pglmm\_ARMA isn't optimized. I then applied the GLS part of remotePARTS, estimating the range parameter from the residuals of the time series analyses. To simplify the comparison, speed computations, and deal with the really small map (which has a large effect by generating high correlations between the estimates of the nugget and the range0, I simulated the data with nugget = 0 and assumed nugget = 0 for fitting both pglmm\_ARMA and the GLS. I performed 1000 simulations for z = 0, 0.25, 0.5, and 0.75 to give both type I errors and power. The value of range in the simulations was 0.2. I also included moderate temporal autocorrelation, with b = 0.5 for all pixels.

The results are in the table below. The columns for "reject" are the proportions of simulated datasets in which H0:z = 0 was rejected at a 0.05 alpha significance level. The mean estimates of z and range are also given.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| simulated c | reject.glmm | reject.gls | c.glmm | c.gls | range.glmm | range.gls |
| 0 | 0.033 | 0.042 | -0.002 | -0.001 | 0.158 | 0.151 |
| 0.25 | 0.157 | 0.172 | 0.242 | 0.239 | 0.157 | 0.152 |
| 0.5 | 0.612 | 0.612 | 0.494 | 0.492 | 0.157 | 0.151 |
| 0.75 | 0.919 | 0.916 | 0.736 | 0.736 | 0.159 | 0.151 |

**Conclusions**

The similarities between methods are surprisingly close for this simple simulation comparison. Even though the GLS is performing all of the evaluation of significance based only on the pixel-level estimates of the time trends, it has the same power as the full lmm that analyzing the spatiotemporal data. This is all the more surprising because the map is so small. The time series are length 30, and the map has 64 pixels, so the temporal dimension is only slightly less than the spatial dimension. There is a lot of information that the GLS doesn't use.

Both methods underestimate the range, but for these small maps, this doesn't inflate type I errors when H0:z = 0 is true. I fit pglmm\_ARMA with ML, although I could have done this with REML which might have had slightly better performance.

It might be interesting to figure out the asymptotic properties of the PARTS approach in a rigorous fashion, but that could be hard.

It might be useful to include this in Clay's simulation manuscript, since it gives the comparison with a gold standard. I also initially thought to suggest including one more simulation. Right now there is a simulation of

X(t) = l0 + (b0 + b1L)t + e(t)

e(t) = r e(t-1) + d(t)

This could be expanded to contain a random component of the pure spatial variation:

X(t) = l0 + (b0 + b1L + b)t + e(t)

e(t) = r e(t-1) + d(t)

where b is a random spatial variable. But then I thought this was a bigger statistical issue that we could address in AIST2.

**LMM vs. PARTS 28Jan21**

I thought I better check PARTS against a "gold standard". As a gold standard, I'm using the model that Mike Bosch is using for his aphid suction trap analyses; he was doing this last spring. In brief, it is a LMM in which the residuals are ARMA(p,q); it is modified from the function pglmm in the phyr package in R that is designed for phylogenetic analyses, so we've called it pglmm\_ARMA(). The residuals also have spatial autocorrelation in the same form as the simulation model in the RSE manuscript. Specifically, the model is

*xi*(*t*) = a*i* + z*it* + e*i*(*t*)

e*i*(*t*) = b*i*e*i*(*t*–1) + d*i*(*t*) (Eq. 1)

where *xi*(*t*) is the value of interest (e.g., NDVI) in pixel (location) *i* in year *t*, a*i* is the intercept, and z*i* is a coefficient that measures the effect of time *t* on *xi*(*t*), where *t* = 1, 2, ..., *T*. Environmental variation that affects changes in *xi*(*t*) from one year to the next is given by the random variable e*i*(*t*). To account for possible temporal autocorrelation, e*i*(*t*) is governed by a normal (Gaussian) autoregressive process generated from the normal random variable d*i*(*t*) that has mean zero and variance s2, with values independent through time so that E{d*i*(*t*) d*i*(*s*)} = 0 for *s* ≠ *t*. The dependence of e*i*(*t*) on b*i* e*i*(*t*–1) generates temporal autocorrelation.

The issue that made me decide to look into this is the fact that the PARTS analyses of Alaska and the continents have standard errors of time trends c that are small compared to the sd of the fitted GLS model (see the new text in the RSE manuscript highlighted in yellow). This is equivalent to having z*i* in the simulation model vary among pixels *i*. Therefore, I modified the simulation model here (but not in the RSE manuscript) to have z*i* follow a Gaussian distribution with mean z*i* and standard deviation z*i*.sd.

For the model (and fitting), the temporal component of the residuals are just AR(1), although pglmm\_ARMA() can take arbitrary *p* and *q*. pglmm\_ARMA() can also include random effects such as z*i*. The covariance matrix from the random effects can be specified (e.g., they could be spatial covariance matrices), but here I'm just assuming the random effects have correlation matrices **I** as in the simulations. Thus, the call of the function with random effects for z*i* is

modre <- pglmm\_ARMA(x ~ 1 + time + (1 | site) + (time | site), p = 1, q = 0, distance.matrix = Dist)

Here, the "site" is the pixel, and there are random effects for both the intercept (1|site) and z*i* (time|site). The call without variation in z*i* is

mod <- pglmm\_ARMA(x ~ 1 + time, p = 1, q = 0)

To compare with PARTS, I fit the "regular" model (steps 1-4) and for comparison I used the model in which the nugget was fixed at 0.1. The model with the fitted nugget is similar to the pglmm\_ARMA() model with random effects (time|site). This is because the nugget allows for random (non-spatial) variation in z*i*. The model with fixed nugget doesn't allow the variance in z*i* to change with the data. The PARTS model with estimated nugget had inflated type I errors; the rejection rate was 15% with an alpha of 0.05. Therefore, I just lowered alpha to give rejection rates of ~5%. I wasn't at all surprised by the inflated rejection rates, because the GLS *P*-values that I'm using are conditional on the covariance matrix (i.e., they don't account for uncertainty in the matrix **V**). The simulations were based on a 6x6 map (36 pixels) because the code is slow and not in any way optimized. This makes uncertainty in **V** large; when applying PARTS to real data, this isn't an issue.

The table below gives the results for the cases with the mean of z*i* is 0 and 0.5.

1. Note that the power of mod (the pglmm\_ARMA model without random effects) dropped a lot with increases in the pixel-to-pixel variation in the time trend z*i* (z*i*.sd). The same pattern is seen comparing the GLS in which the nugget is estimated (gls1) to the GLS with fixed nugget (gls).

2. The results for the two pglmm\_ARMA() models (modre and mod) parallel the two GLS models (gls1 and gls); the parallel is so close it surprised me.

3. Although the comparison isn't completely fair since modre had deflated type I errors, the power of gls1 and modre are very similar. I was actually surprised by this, since I thought the power of gls1 would be considerably less.

4. The parameters fit by modre are pretty good. The estimate of z*i*.sd is the random effect time.site. The model also nails the temporal autocorrelation coefficient, *b*, that was set to 0.5 in the simulations.

5. The random variation in z*i* has the interesting effect of increasing the estimate of *b* in mod. This makes sense. Because there are fixed spatial differences in z*i,* and because mod can't directly handle them, instead mod puts the variation in z*i* into the temporal autocorrelation. This has the effect of allowing the sites (pixels) to differ from each other.

**Conclusions**

a. Fixed spatial variation in the time trend is potentially a problem. However, because PARTS estimates a nugget, it can account for this.

These simulations assumed that the fixed differences in time trends among pixels did not have spatial autocorrelation. This raises the issue of whether the spatial autocorrelation (range) estimated from the residuals of the pixel-level time-series analyses is correct, because it wouldn't pick up spatial autocorrelation in the fixed differences in pixel-level time trends. In PARTS, it is also possible to estimate the spatial range in the GLS fitting, and for the data analyzed in the RSE manuscript, this doesn't seem to be a problem.

b. PARTS (GLS) seems to have pretty good power in comparison to the "gold standard".

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | P-value for H0:z*i* = 0 |  |  |  | nugget |  |  |  | mod vs modre | | *b* |  |
| simulated  z*i* | z*i*.sd | mod | modre | gls | gls1 | mod | modre | gls | gls1 | delta AIC | time.site | mod | modre |
| 0 | 0 | 0.02 | 0.03 | 0.08 | 0.06 | 0.10 | 0.10 | 0.10 | 0.30 | 3.25 | 0.04 | 0.53 | 0.52 |
|  | 0.1 | 0.03 | 0.04 | 0.07 | 0.07 | 0.10 | 0.10 | 0.10 | 0.33 | 2.06 | 0.08 | 0.54 | 0.52 |
|  | 0.2 | 0.02 | 0.02 | 0.06 | 0.08 | 0.11 | 0.10 | 0.10 | 0.38 | -1.87 | 0.17 | 0.56 | 0.52 |
|  | 0.3 | 0.03 | 0.04 | 0.05 | 0.08 | 0.12 | 0.10 | 0.10 | 0.46 | -12.74 | 0.30 | 0.59 | 0.52 |
|  | 0.4 | 0.03 | 0.05 | 0.03 | 0.06 | 0.12 | 0.10 | 0.10 | 0.53 | -26.73 | 0.41 | 0.62 | 0.51 |
|  | 0.5 | 0.01 | 0.05 | 0.01 | 0.08 | 0.13 | 0.10 | 0.10 | 0.62 | -45.24 | 0.52 | 0.66 | 0.51 |
|  | 0.6 | 0.00 | 0.04 | 0.01 | 0.08 | 0.14 | 0.10 | 0.10 | 0.67 | -62.51 | 0.61 | 0.69 | 0.51 |
|  | 0.7 | 0.01 | 0.06 | 0.01 | 0.08 | 0.14 | 0.09 | 0.10 | 0.71 | -80.04 | 0.71 | 0.72 | 0.51 |
|  | 0.8 | 0.01 | 0.04 | 0.00 | 0.10 | 0.13 | 0.09 | 0.10 | 0.74 | -96.40 | 0.80 | 0.75 | 0.51 |
|  | 0.9 | 0.00 | 0.04 | 0.01 | 0.05 | 0.13 | 0.09 | 0.10 | 0.80 | -113.24 | 0.90 | 0.78 | 0.51 |
|  | 1 | 0.00 | 0.04 | 0.01 | 0.06 | 0.14 | 0.09 | 0.10 | 0.83 | -125.03 | 0.98 | 0.80 | 0.51 |
| 0.5 | 0 | 0.41 | 0.43 | 0.58 | 0.42 | 0.10 | 0.10 | 0.10 | 0.31 | 3.20 | 0.04 | 0.53 | 0.52 |
|  | 0.1 | 0.41 | 0.44 | 0.53 | 0.37 | 0.10 | 0.10 | 0.10 | 0.32 | 2.22 | 0.07 | 0.54 | 0.52 |
|  | 0.2 | 0.39 | 0.46 | 0.51 | 0.39 | 0.11 | 0.10 | 0.10 | 0.40 | -2.30 | 0.17 | 0.56 | 0.52 |
|  | 0.3 | 0.35 | 0.45 | 0.44 | 0.38 | 0.12 | 0.10 | 0.10 | 0.47 | -12.38 | 0.29 | 0.59 | 0.52 |
|  | 0.4 | 0.31 | 0.43 | 0.36 | 0.34 | 0.12 | 0.09 | 0.10 | 0.51 | -26.70 | 0.40 | 0.62 | 0.51 |
|  | 0.5 | 0.26 | 0.46 | 0.29 | 0.38 | 0.13 | 0.10 | 0.10 | 0.63 | -45.25 | 0.52 | 0.66 | 0.51 |
|  | 0.6 | 0.19 | 0.46 | 0.19 | 0.38 | 0.13 | 0.09 | 0.10 | 0.67 | -63.57 | 0.62 | 0.69 | 0.51 |
|  | 0.7 | 0.19 | 0.43 | 0.18 | 0.34 | 0.14 | 0.09 | 0.10 | 0.73 | -79.81 | 0.70 | 0.72 | 0.51 |
|  | 0.8 | 0.14 | 0.38 | 0.12 | 0.35 | 0.14 | 0.09 | 0.10 | 0.77 | -96.32 | 0.80 | 0.75 | 0.51 |
|  | 0.9 | 0.12 | 0.40 | 0.10 | 0.29 | 0.15 | 0.10 | 0.10 | 0.79 | -113.58 | 0.90 | 0.78 | 0.50 |
|  | 1 | 0.07 | 0.38 | 0.05 | 0.33 | 0.15 | 0.10 | 0.10 | 0.83 | -127.60 | 0.99 | 0.81 | 0.51 |