STAT 510: Spatiotemporal Stats

Exploratory Modeling of SPT Data (Part 2)

Prof. Taylor-Rodriguez

Trend-Surface Estimation

Trend-Surface Estimation

An alternative to doing prediction based on deterministic methods is to use simple statistical models

- The idea is to try to capture all ST dependence in the *trend*

So what is gained by doing this?

- Easily implementable
- Provides model based error estimate
- Provides model based prediction-error variance
- We can also use cv to assess performance

Trend-Surface Estimation

For simplicity assume we have all locations $\{\mathbf{s}_1, \ldots, \mathbf{s}_m\}$ measured at all time points $\{t_1, \ldots, t_T\}$, such that

$$Z(\mathbf{s}_i; t_j) = \beta_0 + \beta_1 X_1(\mathbf{s}_i; t_j) + \cdots + +\beta_1 X_p(\mathbf{s}_i; t_j) + \epsilon(\mathbf{s}_i; t_j),$$

- $\epsilon(\mathbf{s}_i; t_j) \overset{iid}{\sim} N(0, \sigma^2)$.
- $X_j(\cdot;\cdot)$'s represent spatially varying, temporally varying, and/or spatio-temporally varying predictors
- could also represent ST *basis functions*

Basis Functions

Under certain regularity conditions, it is possible to decompose curves or surfaces using a linear combination of *elemental basis functions*.

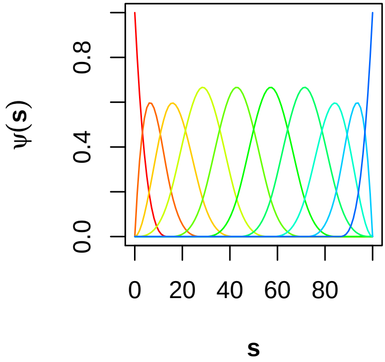
For example, a surface $Y(\mathbf{s})$ in space can be represented as

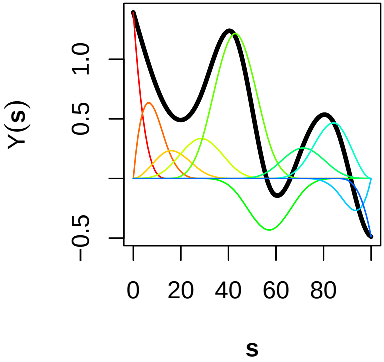
$$Y(\mathbf{s}) = \alpha_1\phi_1(\mathbf{s}) + \alpha_2\phi_2(\mathbf{s}) + \cdots + \alpha_r\phi_r(\mathbf{s})$$

- $\{\phi_k(\mathbf{s})\}$ denoting a **known** set of basis functions (can have local or global support)
- $\{\alpha_k\}$ represent constants that weight the relative importance of each basis function

Note here the absence of error, we are not dealing with data but with the *process* function

Basis Functions





Basis Functions

Some examples of basis functions are

polynomials, splines, wavelets, sines and cosines

If $Y(\mathbf{s})$ is a random process, a statistical model would assume known basis functions $\{\phi_k(\mathbf{s})\}$ and random weights $\{\alpha_k\}$, with a data model, for example, given by

$$\begin{aligned} Z(\mathbf{s}) &= Y(\mathbf{s}) + \epsilon(\mathbf{s}) \\ &= \alpha_1\phi_1(\mathbf{s}) + \alpha_2\phi_2(\mathbf{s}) + \cdots + \alpha_r\phi_r(\mathbf{s}) + \epsilon(\mathbf{s}) \end{aligned}$$

Very cool... these models are easy to fit and can be super flexible

Trend-Surface Estimation: Example

Consider the NOAA daily Tmax data for July of 1993, which has $m = 138$ locations, each measured every day of the month (i.e., $T = 31$). Let's use as covariates:

$X_0(\cdot;\cdot) = 1$: Intercept
$X_1(\cdot;\cdot)$: lon
$X_2(\cdot;\cdot)$: lat
$X_3(\cdot;\cdot)$: t
$X_4(\cdot;\cdot)$: lon$\times$lat

$X_5(\cdot;\cdot)$: lon$\times$t
$X_6(\cdot;\cdot)$: lat$\times$t
$X_k(\cdot;\cdot) = \phi_{k-6}(\cdots)$: with $k = 7, \ldots, 18$ spatial-only basis functions

Trend-Surface Estimation:

Now, let's fit the model

$$Z(\mathbf{s}_i;t_j) = \beta_0 + \beta_1 X_1(\mathbf{s}_i;t_j) + \cdots + \beta_{18} X_{18}(\mathbf{s}_i;t_j) + \epsilon(\mathbf{s}_i;t_j),$$

using *ordinary least squares*

$$RSS = \sum_{j=1}^{T}\sum_{i=1}^{m}(Z(\mathbf{s}_i;t_j) - \hat{Z}(\mathbf{s}_i;t_j))^2$$

to find parameter estimates

$$\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \cdots, \hat{\beta}_{18})' = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Z},$$

with $\text{var}(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}'\mathbf{X})^{-1} \approx \hat{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1}$,

Trend-Surface Estimation: Fitting the model

Let's make the spatial basis fns with FRK::auto_basis()

```
G <- auto_basis(data = (Tmax_long[,c("lon","lat")] %>%
                        SpatialPoints()), # make Tmax a spp object
                nres = 1,
                type = "Gaussian")
```

Evaluate basis fns at locations of interest

```
coords <- as.matrix(Tmax_long[,c("lon","lat")])
S <- eval_basis(basis = G,    # basis functions
                s =  coords) %>%   # eval at these locations
                as.matrix()                # conv. to matrix
colnames(S) <- paste0("B", 1:ncol(S))

Tmax2 <- cbind(Tmax_long, S) %>%
  dplyr::select(-year,-month,-proc,-julian,date)
```

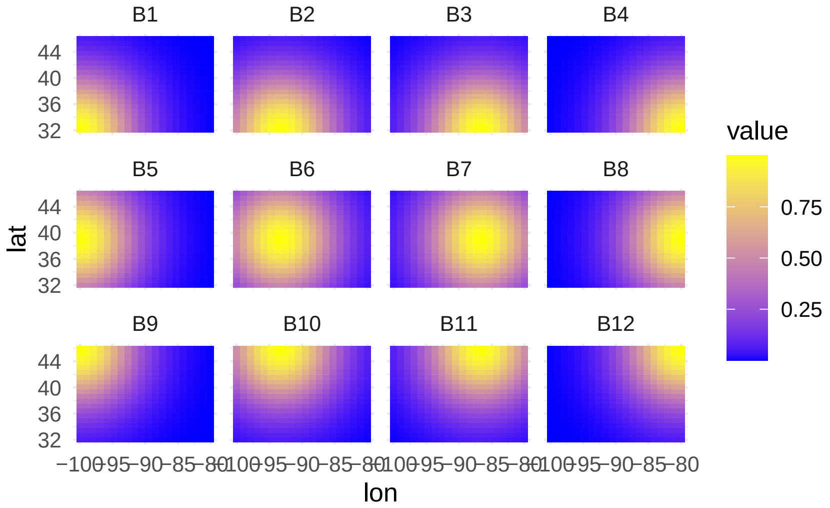
Fit the model

```
#remove the 14th
Tmax_no_14 <- filter(Tmax2, !(day == 14))

Tmax_July_lm <- lm(z ~ (lon + lat + day)^2 + .,
                   data = dplyr::select(Tmax_no_14,
                                        -id,-t,-date))
```

Trend-Surface Estimation: Smoothing

Let's generate prediction grid



Trend-Surface Estimation: Smoothing

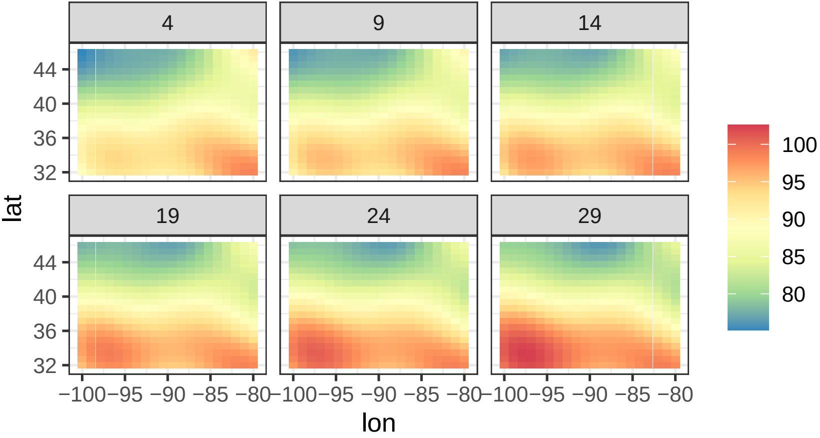
```r
pred_grid <- expand.grid(lon = seq(-100, -80, length = 20),
                         lat = seq(32, 46, length = 20),
                         day = seq(4,31,by=5))
# generate basis fns at prediction points
S.pred <- eval_basis(basis = G,
                     s =  as.matrix(dplyr::select(pred_grid,lon,lat))) %>%
  as.matrix()
colnames(S.pred) <- paste0("B", 1:ncol(S.pred))

# append to basis prediction grid
pred_grid <- cbind(pred_grid,S.pred)
```

```r
#gte predictions including 95% pred int.
preds <- predict(Tmax_July_lm,
                 newdata=pred_grid,
                 interval = "prediction")
pred_grid <- pred_grid %>%
  bind_cols(as_tibble(preds))
```

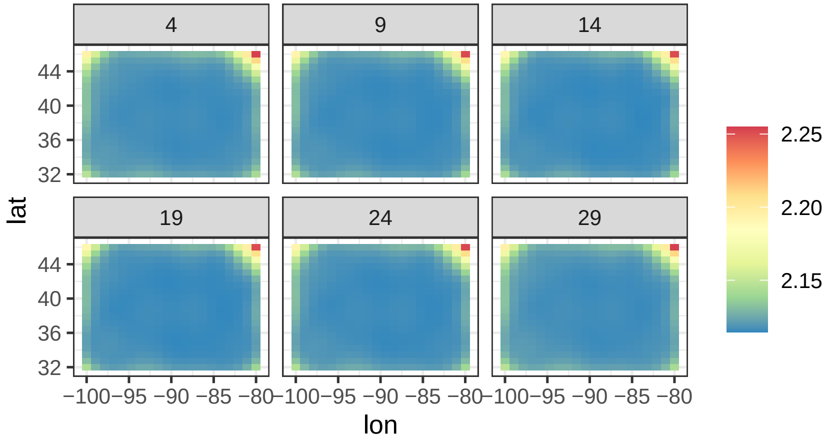
Trend-Surface Estimation: Smoothing

Fitted Values



Trend-Surface Estimation: Smoothing

Standard Errors



Trend-Surface Estimation: Smoothing - Comments

- Prediction SE's don't show structure, but display undertainty increasing at domain boundaries (extrapolation)
- Predictions smoother than kernel predictions from before (due to smooth basis fns), but it's not always the case depends on predictors
- This model does not explicitly account for response measurement errors
- Variation from measurement error confounded with variation due to lack of fit
- Note that the regression predictor can be considered a type of kernel predictor

Trend-Surface Estimation: Parameter Estimation

Table 1:

	Dependent variable:
	z
lon	1.757 (1.088)
lat	−1.317 (2.556)
day	-1.216^{***} (0.134)
B1	16.647^{***} (4.832)
B2	18.528^{***} (3.056)
B3	-6.607^{**} (3.172)
B4	30.545^{***} (4.370)
B5	14.739^{***} (2.747)
B6	-17.541^{***} (3.423)
B7	28.472^{***} (3.552)
B8	-27.348^{***} (3.164)
B9	-10.235^{**} (4.457)
B10	10.558^{***} (3.327)
B11	-22.758^{***} (3.533)
B12	21.864^{***} (4.813)
lon:lat	−0.026 (0.028)
lon:day	-0.023^{***} (0.001)
lat:day	-0.019^{***} (0.002)
Constant	192.243^{**} (97.854)
Observations	3,989
R^2	0.702
Adjusted R^2	0.701
Residual Std. Error	4.225 (df = 3970)
F Statistic	520.410^{***} (df = 18; 3970)
Note:	*p<0.1; **p<0.05; ***p<0.01

Trend-Surface Estimation: Diagnostics

After fitting a model such as this, we need to check:

- non-constant error variance
- error (in)dependence (specially important with ST data)
- outliers and influential observations
- multicollinearity
- non-normality, etc. . .

Trend-Surface Estimation: Diagnostics
Error dependence checks

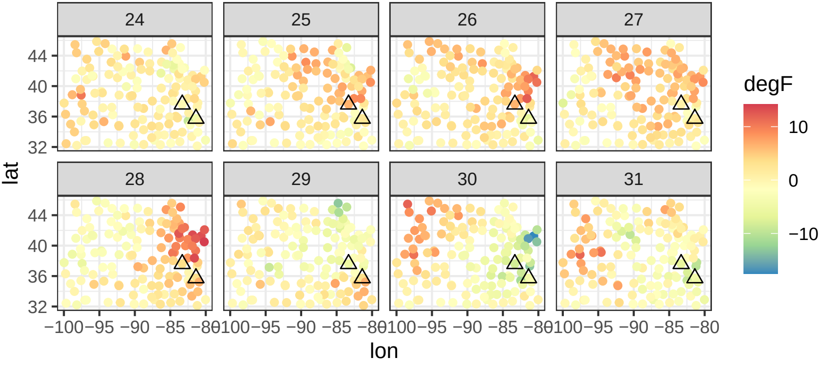
```
library(modelr)
Tmax_no_14 <- Tmax_no_14 %>%
  add_predictions(Tmax_July_lm) %>%
  add_residuals(Tmax_July_lm)

#plotting spatial residuals last 7 days
ggplot(filter(Tmax_no_14, day %in% 24:31)) +
  geom_point(aes(lon, lat, colour = resid)) +
  facet_wrap(~ day, ncol=4) +
  col_scale(name = "degF") +
  geom_point(data = filter(Tmax_no_14,day %in% 24:31 &
                               id %in% c(3810, 3889)),
             aes(lon, lat), colour = "black",
             pch = 2, size = 2.5) +
  theme_bw()

#plotting temporal residuals 2 stations
Tmax_no_14 %>%
  filter(id %in% c(3810, 3889)) %>%
  mutate(id=as.character(id)) %>%
  ggplot(aes(x=day,y=resid)) +
  geom_line(aes(group=id,colour=id)) +
  theme_bw()
```

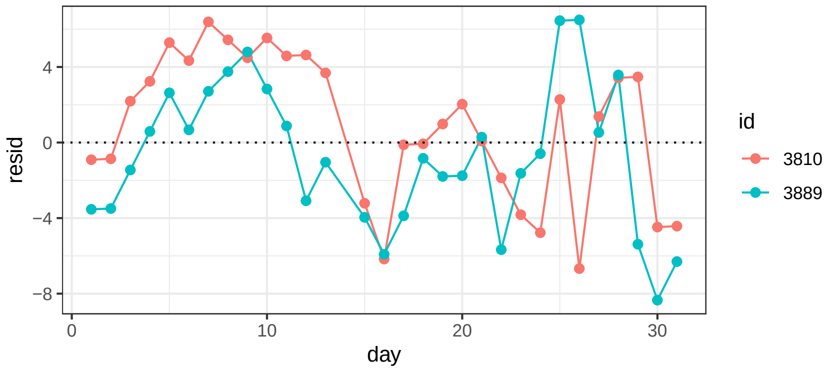
Trend-Surface Estimation: Diagnostics

Residual Analysis: Spatial Residuals



Trend-Surface Estimation: Diagnostics

Residual Analysis: Temporal Residuals



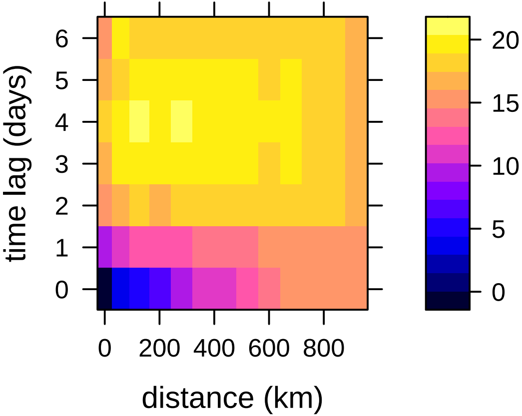
Trend-Surface Estimation: Diagnostics

Error dependence checks (Semivariogram)

$$F = \left| \frac{\hat{\gamma}_e(\|\mathbf{h}_1\|; \tau_1)}{\hat{\sigma}^2} - 1 \right|$$

$\mathbf{h}_1$ and τ_1 denote smallest possible lags in space and time, respectively.

reject if F large (determine what is *large* permuting ST locations)



Trend-Surface Estimation: Diagnostics

Error dependence checks

Durbin-Watson Statistic

$$d = \frac{\sum_{t=2}^{T}(\hat{e}_t - \hat{e}_{t-1})^2}{\sum_{t=1}^{T}\hat{e}_t^2}, \quad d \in [0, 4]$$

$d = 2 \Rightarrow$Indep, $d \to 0 \Rightarrow (+\text{dep})$, $d \to 4 \Rightarrow (-\text{dep})$

Moran's I

$$I = \frac{m \sum_{i=1}^{m}\sum_{j=1}^{m} w_{ij}(Z_i - \bar{Z})(Z_i - \bar{Z})}{\left(\sum_{i=1}^{m}\sum_{j=1}^{m} w_{ij}\right)\sum_{i=1}^{m}\sum_{j=1}^{m}(Z_i - \bar{Z})^2}$$

Trend-Surface Estimation: Diagnostics

Effect of error dependence

On the bright side

- Regression coefficients and predictions are still unbiased

The not so great side

- SE's and prediction SE's wrong, for ST data usually underestimated!!!

How can we fix it?

- Modeling error dependence: assume $\mathbf{e} \sim \mathbf{N_{mT}}(\mathbf{0}, \mathbf{C_e})$
- What challenges do you see with this fix? (more on Lab 02 and in Ch 4)

In-class Exercise

Under ST dependence, $\mathbf{Z} \sim N_{mT}(\mathbf{X}\beta, \mathbf{C}_e)$. Note that the likelihood for $\mathbf{Z}$ is maximized when minimizing

$$(Z - \mathbf{X}\beta)' C_e^{-1} (Z - \mathbf{X}\beta).$$

Derive the Generalized Least Squares Estimator for β

Trend-Surface Estimation: Variable Selection

Options

- Best subsets (with `leaps` function)
- Stepwise and forward selection (with `step` function)
- Penalized regression (e.g., Ridge and Lasso)

Trend-Surface Estimation: Variable Selection

Penalized Regression

$$Z(\mathbf{s}_i; t_j) = \underbrace{\beta_0 + \beta_1 X_1(\mathbf{s}_i; t_j) + \cdots + \beta_{18} X_{18}(\mathbf{s}_i; t_j)}_{=\hat{Z}(\mathbf{s}_i; t_j)} + \epsilon(\mathbf{s}_i; t_j),$$

OLS estimates β by minimizing

$$RSS = \sum_{j=1}^{T} \sum_{i=1}^{m} (Z(\mathbf{s}_i; t_j) - \hat{Z}(\mathbf{s}_i; t_j))^2$$

Penalized approaches estimate β by minimizing

$$RSS + \lambda \sum_{j=1}^{p} |\beta_j|^q$$

Ridge: $q = 1$, Lasso: $q = 2$