More covariance and Gaussian processes

ISTA 410 / INFO 510: Bayesian Modeling and Inference

U. of Arizona School of Information

April 14, 2021

Outline

Last time:

- Varying effects models; covariance between intercepts and slopes
- Multivariate normal distributions and covariance matrices

with LKJ prior

Today:

- Covariance that varies with space or time
- Gaussian processes

Example: political trends

US presidential voting and age

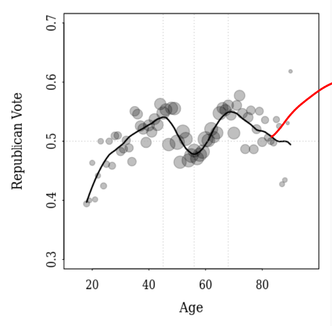
Example due to Gelman and Ghitza:

- Generational model of partisan preferences
- Model influence of political events on preference

US presidential voting and age

Example due to Gelman and Ghitza:

- Generational model of partisan preferences
- Model influence of political events on preference



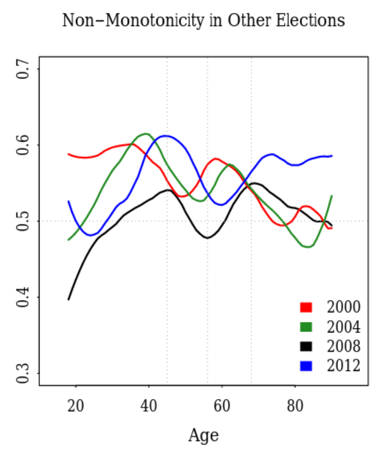
local regression (LOESS)

(2008 US pres. election)

It's a cliche in US politics that older voters are more conservative

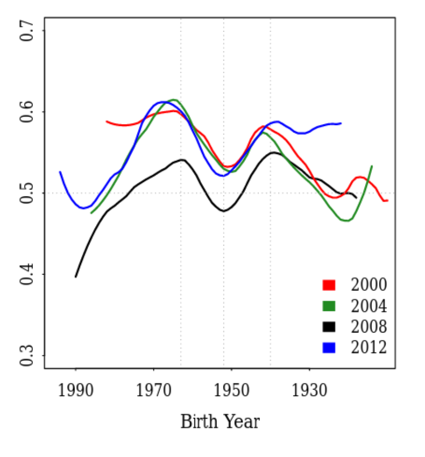
US presidential voting and age

What about other elections?



US presidential voting and age

What about other elections?



Modeling the birth year effect

Survey respondents (voters) binned into groups by

- Birth year
- Year of election observed
- race/region $\in$ {minority, Southern white, non-Southern white}

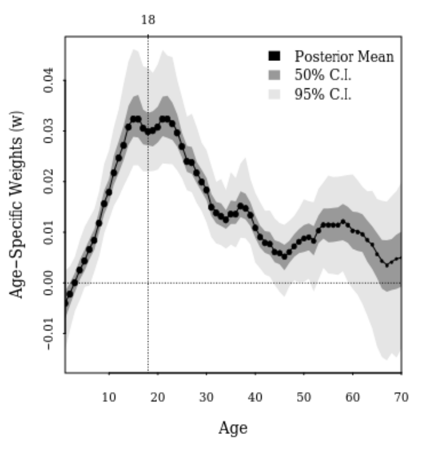
varying intercepts

Used to estimate "age weights": how much the political situation at a given age influences

Priors set on age weights to enforce similarity among nearby ages

Predictor for varying slopes: $x_{i,j} \to$ i (age), j (born) $x_{i,j}$ – Republican approval.

Age effects


18
Posterior Mean
50% C.I.
95% C.I.
Age−Specific Weights (w)
−0.01
0.00
0.01
0.02
0.03
0.04
10
20
30
40
50
60
70
Age

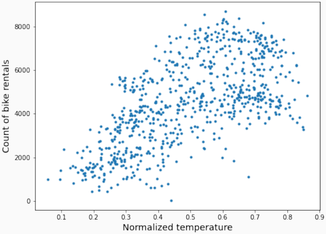
Example: bike share data

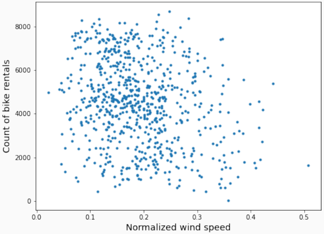
Bike share programs

- Bike share programs: short-term rentals for bicycles
- Have data on count of renters, along with daily weather data

Goal: estimate influence of temperature, wind speed

Make a plot to check reasonableness





Simple Poisson regression model

We have count data, so use Poisson regression:

rate / expected count on day j

$$y_j \sim \text{Poisson}(\boxed{\lambda_j})$$

$$\underbrace{\log}_{\text{log link function}} \lambda_j = \underbrace{\alpha + \beta_T T_j + \beta_w w_j}_{\text{linear model for log rate}}$$

$$\alpha \sim \text{Normal}(0, 5)$$

$$\beta_T \sim \text{Normal}(0, 1)$$

$$\beta_w \sim \text{Normal}(0, 1)$$

logit → binomial

log → Poisson

$$\text{logit}(x) = \log \frac{x}{1-x}$$

Results and predictive check

- Summary:

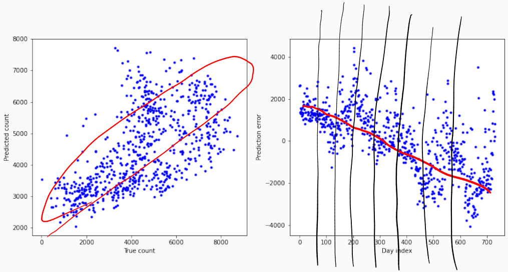
	mean	sd	hdi_3%	hdi_97%
alpha	7.812	0.002	7.808	7.817
beta_temp	1.450	0.003	1.444	1.456
beta_wind	-0.823	0.008	-0.837	-0.809

Results and predictive check

- Summary:

	mean	sd	hdi_3%	hdi_97%
alpha	7.812	0.002	7.808	7.817
beta_temp	1.450	0.003	1.444	1.456
beta_wind	-0.823	0.008	-0.837	-0.809

- Posterior predictive error vs. date:



Adding in varying intercepts

The posterior predictions show mediocre fit to data – in particular, prediction error clearly follows a trend over time.

Add in varying intercepts by month:

$$
\begin{aligned}
y_j &\sim \mathrm{Poisson}(\lambda_j) \\
\log \lambda_j &= \alpha_{\mathrm{month}(j)} + \beta_T T_j + \beta_w w_j \\
\beta_T &\sim \mathrm{Normal}(0, 1) \\
\beta_w &\sim \mathrm{Normal}(0, 1) \\
\alpha_{\mathrm{month}(j)} &\sim ?
\end{aligned}
$$

Varying intercepts

We could simply use our usual strategy and do something like:

$$\alpha \sim \mathrm{Normal}(\mu, \tau)$$

with some hyperpriors on μ, τ

- Usual multilevel strategy oriented around the idea of exchangeable groups
- Share information among groups
- Exchangeability: the model doesn't change if we permute the index of the groups
- Time points not really exchangeable

Varying intercepts

Alternative:

- Sample varying intercepts from a multivariate normal with correlations
- Here, we can impose some structure on the correlations:
  - Months closer in time are more similar
  - Months closer in time should have higher correlations
- How do we impose this? Put it into the covariance matrix

New model

Make α a multivariate normal:

$$\boldsymbol{\alpha} \sim \text{MVNormal}\left(\overbrace{\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}}^{\text{24 components}}, \boxed{\mathsf{K}} \right)$$

24 x 24 matrix

- Covariance between α_i, α_j should depend on how close months i and j are in time
- So, K_{ij} should be a function of i, j

Covariance function

Set:

$$\mathsf{K}_{ij} = \eta^2 \exp\left(-\frac{(i-j)^2}{2\ell^2}\right) + \underline{\underline{\sigma^2\delta_{ij}}}$$

$$D = \begin{pmatrix} 0 & 1 & 2 & 3 & & & 23 \\ 1 & 0 & 1 & 2 & 3 & & \\ 2 & 1 & 0 & 1 & 2 & & \\ 3 & 2 & 1 & 0 & 1 & \ddots & \\ & 3 & 2 & 1 & \ddots & \ddots & \\ & & \ddots & \ddots & \ddots & & \\ 23 & & & & & & \end{pmatrix}$$

Parameters:

- η^2 – magnitude of correlations
- ℓ^2 – length scale
- σ^2 – self variance
    - Even if your model doesn't need this, a small amount useful for numerical stability

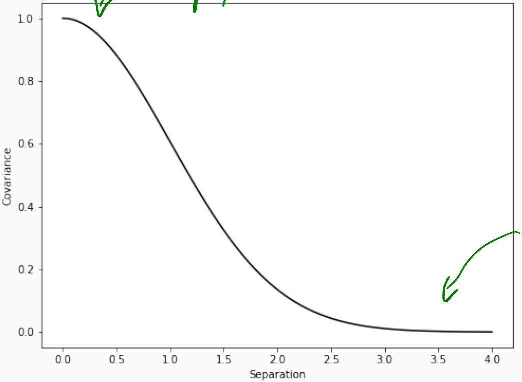
$$\exp(-D^2) = \begin{pmatrix} 1 & e^{-1} & e^{-4} & \cdots \\ & 1 & e^{-1} & \cdots \\ & & 1 & \\ & & & \ddots \end{pmatrix}$$

What does this look like?

What this means is the covariance between two αs is a function of their separation:

$y = \exp(-x^2)$

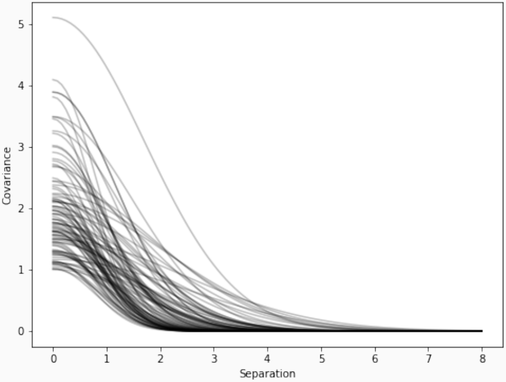
nearby parameters highly correlated



distant parameters are uncorrelated.

What does this look like?

Parameterized by varying η^2, ℓ^2:

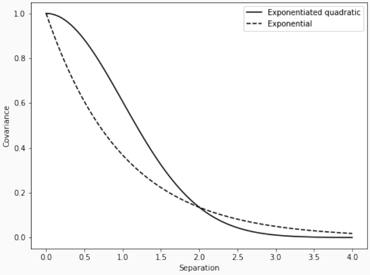


What about a different functional relationship?

The formula from before:

$$\mathsf{K}_{ij} = \eta^2 \exp\left(-\frac{(i-j)^2}{2\ell^2}\right) + \sigma^2\delta_{ij}$$

is an exponentiated quadratic; what about another form?



What to add to the model?

```
with pm.Model() as bike_model:
    beta_temp = pm.Normal('beta_temp', 0, 2)
    beta_wind = pm.Normal('beta_wind', 0, 2)
    eta = pm.Exponential('eta', 1)
    ls = pm.Exponential('ls', 4)

    Kij = (eta ** 2) * pm.math.exp(-(separation ** 2) / (ls ** 2)) + 0.01 * np.
    k = pm.MvNormal('k', mu=tt.zeros(24), cov=Kij, shape = 24)

    theta = pm.math.exp(k[bikes['month_index']] + beta_temp * bikes['temp']
            + beta_wind * bikes['windspeed'])

    y_ = pm.Poisson('y', theta, observed = bikes['cnt'])
```

slope priors

hyperparameters for covariance.

0.01 * np.eye(24)

cov matrix

separation is a 24 × 24 matrix with i, j entry equal to $|i - j|$

model eqn.

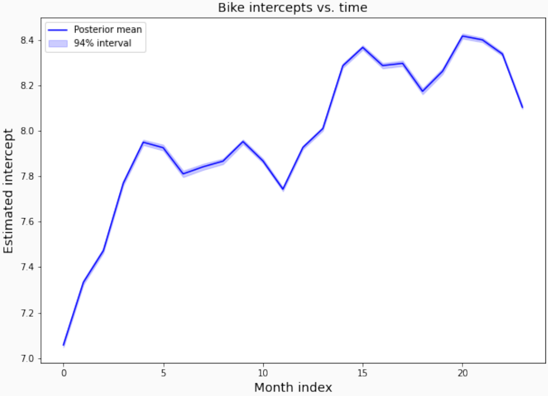
Results

Results from a summary table:

	mean	sd	hdi_3%	hdi_97%
beta_temp	0.965	0.008	0.951	0.982
beta_wind	-0.694	0.008	-0.708	-0.680
alpha[0]	7.058	0.005	7.048	7.069
alpha[1]	7.334	0.005	7.324	7.344
alpha[2]	7.472	0.005	7.463	7.482
alpha[3]	7.769	0.005	7.759	7.779
alpha[4]	7.949	0.006	7.938	7.960

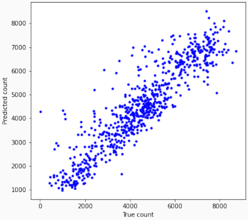
Intercepts over time

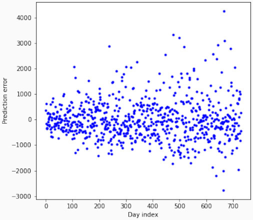
We can look at the intercepts estimated as a function of the month:



Posterior predictive check

Same predictive check as before:





Gaussian processes as random functions

Time grouping in the bike example

In the bike share example:

- We used k_{month} as our varying intercept
- Why monthly?

Time grouping in the bike example

In the bike share example:

- We used k_{month} as our varying intercept
- Why monthly?

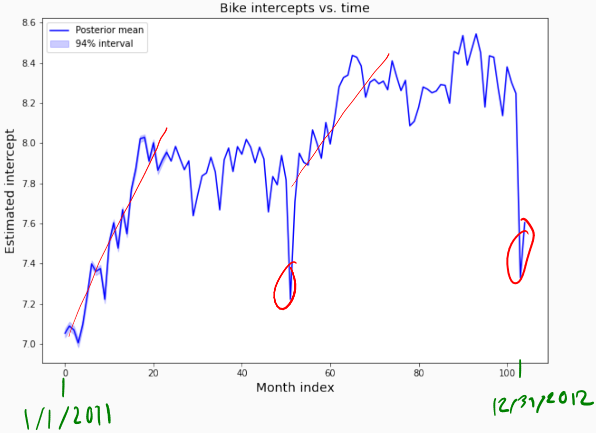
$$\begin{pmatrix} 0 & 1 & 2 & & & 104 \\ & 0 & 1 & 2 & & \\ & & 0 & 1 & 2 & \\ & & & 0 & \ddots & \\ & & & & & \\ 104 & & & & & \end{pmatrix}$$

Try weekly instead:

- 105 varying intercepts
- Same approach: 105-dimensional multivariate normal; covariance matrix built in the same way

Intercepts over time

Now we get more resolution on the intercepts:



Gaussian process regression

- Weekly and monthly versions identical in spirit, just with different data resolution for the intercepts
- Unified way to think of this:

  not a vector of fixed length

  $$\log \lambda_j = \alpha(t_j) + \beta_T T_j + \beta_w w_j$$

  where α is a continuous function of time
- We're not trying to estimate a vector from observations of each component
- We're trying to estimate a function from several observations of function values

GP: the definition

A Gaussian process is a random *function* – i.e., we're really talking about a probability distribution on a space of functions.

The feature that makes a GP a GP: if you pick any n values of x, then the vector of function values $(\mu(x_1), \mu_(x_2), \ldots, \mu(x_n))$ has a multivariate normal distribution:

$$(\mu(x_1), \ldots \mu_(x_n)) \sim \mathrm{Normal}((m(x_1), \ldots, m(x_n)), K(x_1, \ldots, x_n))$$

The GP is determined by its mean function m and covariance K.

GP: the definition

Typically, the covariance matrix is determined by a function called the *kernel* $k(x, x')$.

- $k(x, x')$ determines how much the value of $\mu(x)$ depends on $\mu(x')$.
- Common (not universal) property: $k(x, x')$ depends on the distance between x, x'
- Idea: we're looking for continuous functions, so the values of $\mu(x), \mu(x')$ should be close if x, x' are close; but if they're far apart

Squared exponential covariance

Very common choice: squared exponential covariance function:

could replace with

$$k(x, x') = \eta^2 \exp\left(-\frac{(x - x')^2}{2\ell^2}\right)$$

$\exp\left(-\frac{|x-x'|}{\ell^2}\right)$

Covariance is high when $x - x'$ is small, falls off at longer ranges.

Hyperparameters:

- η: the maximum covariance
- ℓ: the *length scale*, controls how quickly covariance decays

In practice

How this is realized in practice:

- We have a set of observations $f(x_i)$
- GP property says

$$f(x_i) \sim \mathrm{MvNormal}((\overbrace{m(x_1)}^{\text{mean function}}, \ldots, m(x_n)), \overbrace{K(x_1, \ldots, x_n)}^{\text{covariance matrix}})$$

- So we evaluate the covariance function $k(x, x')$ at each pair of observed x values and use that to build a covariance matrix
- The Gaussian process distribution

$$\mathcal{GP}(\mu(\overset{x}{\cdot}), \overset{k(x,x')}{k(\cdot, \cdot')})$$

is really a prior distribution on the space of continuous functions

Summary

Summary:

- Many data sets naturally include observations that should be correlated based on, e.g. time or distance
- Including these correlations amounts to estimating an underlying function
- $\mathcal{GP}$ is a prior distribution on a space of functions, parameterized by a mean function and covariance

Next time:

- More Gaussian process regression
- Various covariance functions