Part 8

Machine learning

ST740

North Carolina State University

What is machine learning?

Bayesian nonparametrics (BNP)

- A parametric analysis assumes a fairly simple data-generating model and learning takes place by estimating the parameters
- The model could be purely statistical, e.g., regression
- The model can also by physical, e.g., an epidemiological (SIR) model
- Advantage: parameters are usually interpretable
- Disadvantage: Inference is invalid and predictions are poor if the model is way off

Bayesian nonparametrics

- A nonparametric analysis attempts to avoid assumptions
- For example, if you want to test if two means are equal, do a rank test instead of assuming normality
- Bayesian methods require a likelihood, so some model must be specified
- BNP specifies models that are very flexible, often with infinitely-many parameters

Bayesian nonparametrics

Consider the polynomial regression model

$$Y_i|\boldsymbol{\theta} \sim \text{Normal}\left(\beta_0 + \sum_{j=1}^{J} X_i^j \beta_j, \sigma^2\right)$$

- Parametric: $J = 1$ or 2 and you need to verify this fits
- Semiparametric: $J = 15$ probably fits almost any function, but you need to tune J
- Nonparametric: $J = \infty$ is the most flexible but requires tricky tricks to implement

Bayesian nonparametrics

- BNP typically replaces priors on parameters with priors on functions
- Example 1: $\mathrm{E}(Y|\mathbf{X}) = \mu(\mathbf{X})$ is a function from $\mathcal{R}^p \to \mathcal{R}^1$
- Gaussian process regression estimates this function assuming only that it is continuous in $\mathbf{X}$
- Example 2: say the errors $\varepsilon_i \sim f$ for some PDF f
- A Dirichlet process mixture of normals prior allows f to be any continuous PDF

Outline

- High-dimensional data
  - **Linear regression**
  - Networks
- Nonparametric regression
  - Generalized additive models
  - Bayesian additive regression trees
  - Gaussian process regression
  - Bayesian deep learning
- Prior for a density function

High-dimensional linear regression

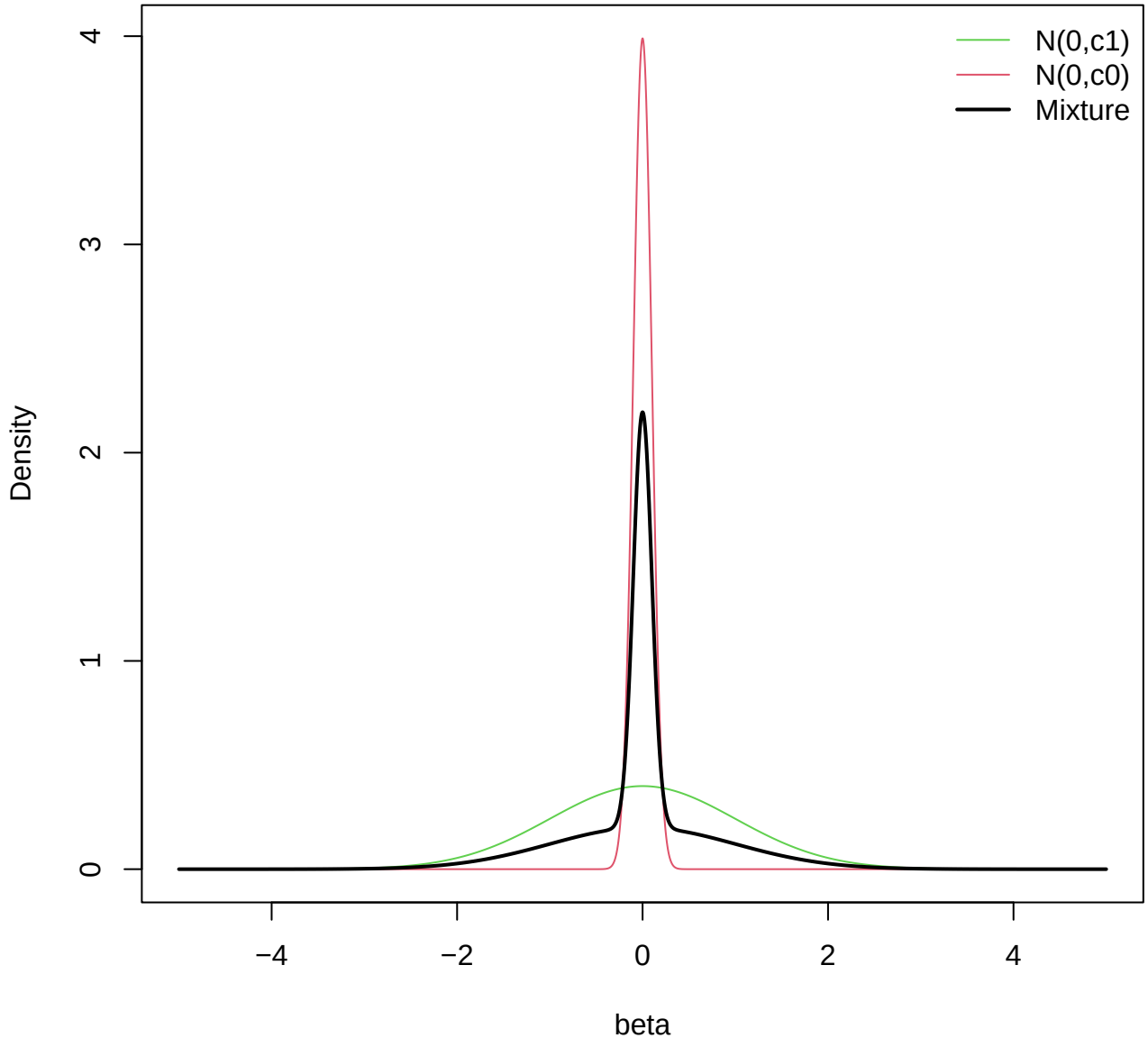
- Consider the linear regression model with $Y_i|\boldsymbol{\beta} \sim \text{Normal}\left(\beta_0 + \sum_{j=1}^{p} X_{ij}\beta_j, \sigma^2\right)$ for $i = 1, ..., n$
- A classical analysis has $p << n$ and the covariates are chosen based on prior scientific knowledge
- A high-dimensional analysis has p large relative to n
- Example, Y_i is the a person's health response and $p = 100,000$ genetic markers
- In this $p >> n$ setting we need new machine learning methods[1]

[1] *Handbook of Bayesian Variable Selection* (2021) by Tadesse and Vannucci provide a recent review

Sparsity priors

- This analysis is impossible without some strong assumptions
- A common assumption is **sparsity**, i.e., most of the β_j are zero
- This assumption is encoded in the prior for the β_j
- A sparsity prior should have mass at or near zero and heavy tails
- This simultaneously shrinks irrelevant variables to zero and reduces bias in the important variables

Spike and slab priors



Spike and slab priors

- The most natural prior is a mixture prior,

$$\pi(\beta) = q\phi(\beta; 0, c_1) + (1 - q)\phi(\beta; 0, c_0)$$

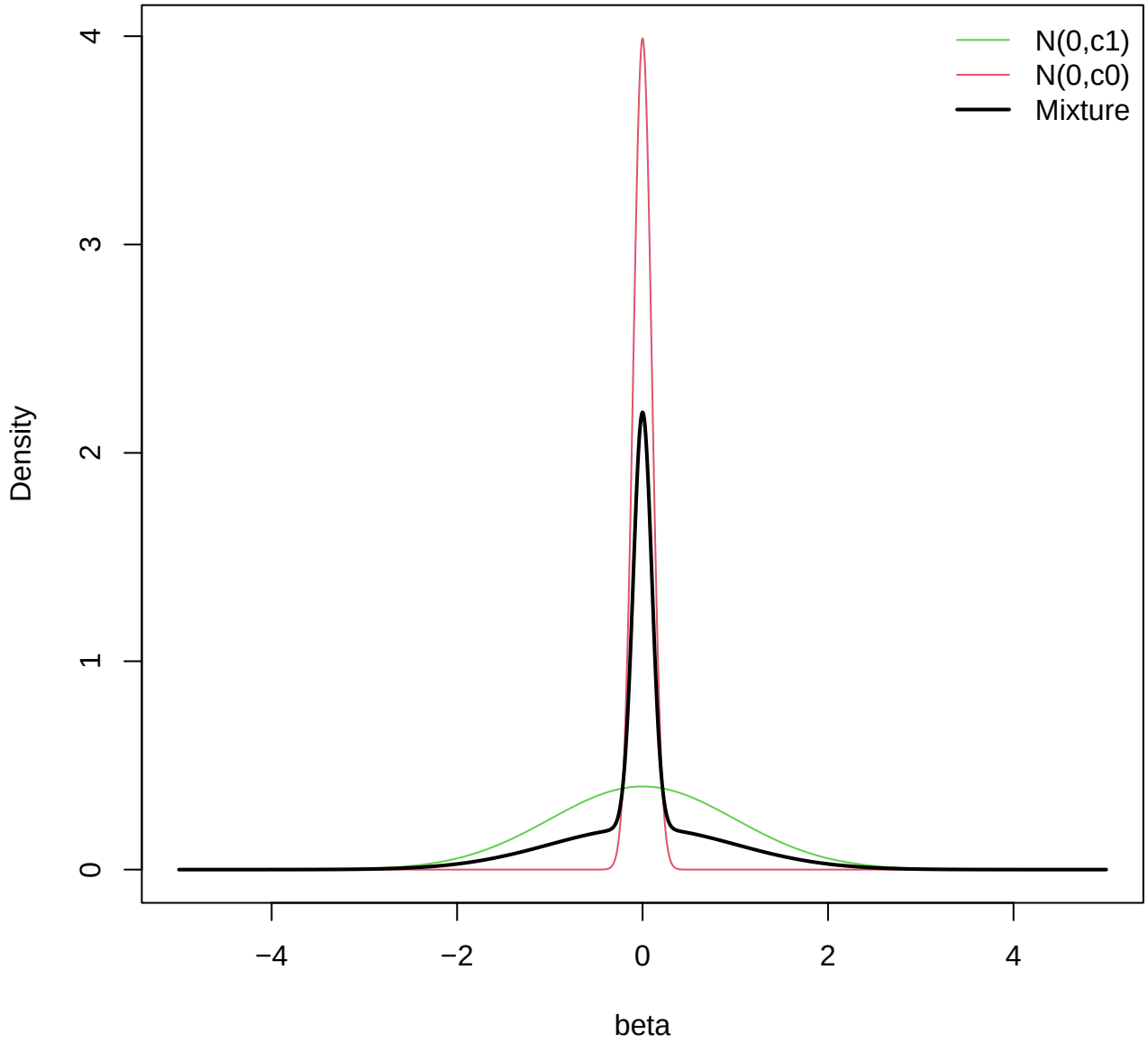
  where $\phi(x; m, s)$ is the Normal(m, s^2) PDF

- The prior probability of inclusion is q
- The prior SD given a variable is included is c_1
- The prior SD given a variable is excluded is $c_0 << c_1$
- Can set $c_0 = 0$ giving a discrete prior

Spike and slab priors

```
c1 <- 1
c0 <- 0.1
q  <- 0.5

beta <- seq(-5,5,.001)

plot(beta,dnorm(beta,0,c0),col=2,type="l",
     ylab="Density")

lines(beta,dnorm(beta,0,c1),col=3)

mix <- q*dnorm(beta,0,c1)+(1-q)*dnorm(beta,0,c0)
lines(beta,mix,lwd=2)

legend("topright",
       c("N(0,c1)","N(0,c0)","Mixture"),
       lwd=c(1,1,2),col=c(3,2,1),bty="n")
```

Spike and slab priors



Spike and slab priors

- Gibbs sampling can be used
- Let $\gamma_j = 1$ if variable j is included and $\gamma_j = 0$ otherwise
- The model is

$$\beta_j | \gamma_j \sim \text{Normal}\left(0, c_{\gamma_j}\right)$$

where $\gamma_j \sim \text{Bernoulli}(q)$

- The full conditional distributions of β_j, γ_j, c_0, c_1 and q are all conjugate
- However, because of the discrete prior on γ_j, convergence can be slow

Spike and slab priors

- The posterior is summarized by the 95% interval for the β_j and inclusion probabilities, Prob($\gamma_j = 1|\mathbf{Y}$)

- You can also compute the most likely model,

$$\gamma = (\gamma_1, ..., \gamma_p)$$

  however estimating model probabilities is hard with large p

- Bayesian model averaging via MCMC is used for prediction

Continuous shrinkage models

- The discrete form of the spike and slab priors slows convergence
- Continuous mixture priors have been proposed as alternatives
- Global-local skrinkage: $\beta|\sigma, \gamma_0, \gamma_j \sim \text{Normal}\left(0, (\sigma\gamma_0\gamma_j)^2\right)$
- Global shrinkage is controlled by γ_0
- Local shrinkage is controlled by $\gamma_j \sim g$ for some mixing distribution g

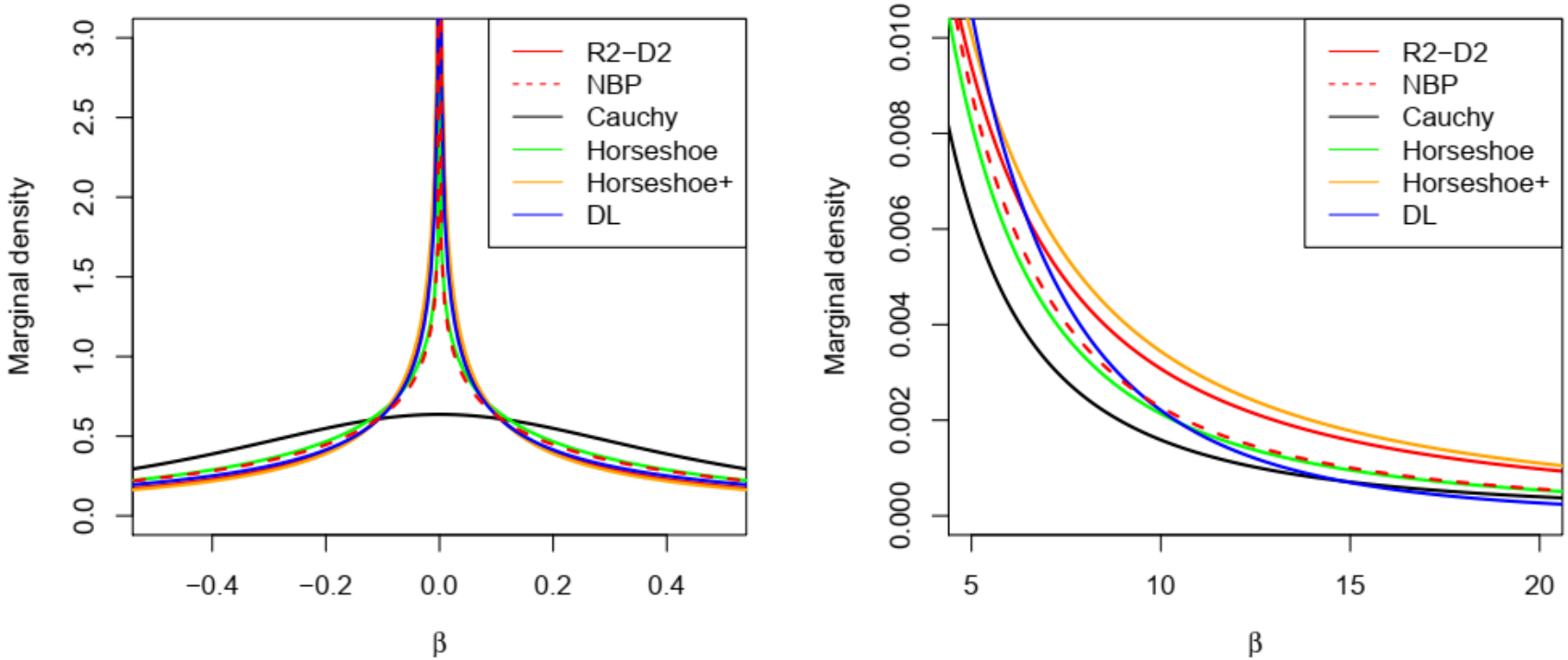
Horseshoe prior

- The horseshoe prior takes g as $\gamma_j \sim$ HalfCauchy
- If $\mathbf{X}$ is the identify matrix and $\gamma_0 = 1$, then

$$\mathrm{E}(\beta_j|\mathbf{Y}) = [1 - \mathrm{E}(\kappa_j|Y_j)]\,Y_j$$

- The shrinkage parameter $\kappa_j = \frac{1}{1+\gamma_j} \sim \mathrm{Beta}(1/2, 1/2)$
- The Beta(1/2,1/2) distribution is shaped like a horseshoe with peaks at 0 and 1
- The induced distribution for β_j (over γ_j) has mass near zero and heavy tails
- Other shrinkage priors have been proposed

Other continuous shrinkage priors



Taken from Zhang et al,
`https://arxiv.org/pdf/1609.00046.pdf`

The R2D2 prior

- In our recent work, we proposed the R2-induced Dirichlet Decomposition (R2-D2) prior,[2]
- The prior places a Beta(a, b) prior on Bayesian R^2
- The proportion of variance allocated to each β_j follows a Dirichlet($c, ..., c$) prior
- Small a promotes skrinkage and small c promotes sparsity
- We (well, Zhang) proved posterior consistency for p increasing faster than n

[2] Zhang, Naughton, Bondell, Reich (2022). Bayesian Regression Using a Prior on the Model Fit: The R2-D2 Shrinkage Prior. *JASA*

Outline

- High-dimensional data
  - Linear regression
  - **Networks**
- Nonparametric regression
  - Generalized additive models
  - Bayesian additive regression trees
  - Gaussian process regression
  - Bayesian deep learning
- Prior for a density function

Gaussian graphical models

- Let $\mathbf{Y}_i = (Y_{i1}, ..., Y_{ip})^T$ be the response for observation $i \in \{1, ..., n\}$
- Rather than a response and predictor, we are interested in learning about the relationships between the p variables
- For example, maybe the p variables are genes and we want to uncover a regulatory network
- This might looks like $Y_{i1} \rightarrow Y_{i6} \rightarrow Y_{i3}$
- Other examples: Neuron firing, social-media influencers, congress, etc

Gaussian graphical models

- A Gaussian model is $\mathbf{Y} \sim \text{Normal}(0, \Sigma)$ for $p \times p$ covariance matrix Σ
- The precision matrix $\Omega = \Sigma^{-1}$ has (i, j) element ω_{ij}
- Y_i and Y_j are correlated conditionally on Y_k for all $k \notin \{i, j\}$ if and only if $\omega_{ij} \neq 0$
- So we could put a sparsity prior on the ω_{ij}

Gaussian graphical models

- Constructing a prior for ω_{ij} is tricky because Ω must be symmetric and positive definite
- Wang[3] shows that the prior below is valid
- The diagonal elements are

$$\omega_{ii} \sim \text{Exponential}$$

- The off-diagonal elements are

$$\omega_{ij} = \omega_{ji} \sim \text{Mixture of Normals}$$

[3] Wang H (2015). Scaling it up: stochastic search structure learning in graphical models. *Bayesian Analysis*

Outline

- High-dimensional data
  - Linear regression
  - Networks
- Nonparametric regression
  - **Generalized additive models**
  - Bayesian additive regression trees
  - Gaussian process regression
  - Bayesian deep learning
- Prior for a density function

Nonparametric regression

- Let $Y_i = \mu(\mathbf{X}_i) + \varepsilon_i$ so that

$$\mathrm{E}(Y_i|\mathbf{X}_i) = \mu(\mathbf{X}_i)$$

- A linear model takes the regression function μ to be linear, $\mu(\mathbf{X}_i) = \beta_0 + \sum_{j=1}^{p} X_{ij}\beta_j$

- In the parametric analysis this goal is to estimate the intepretable parameters β_j

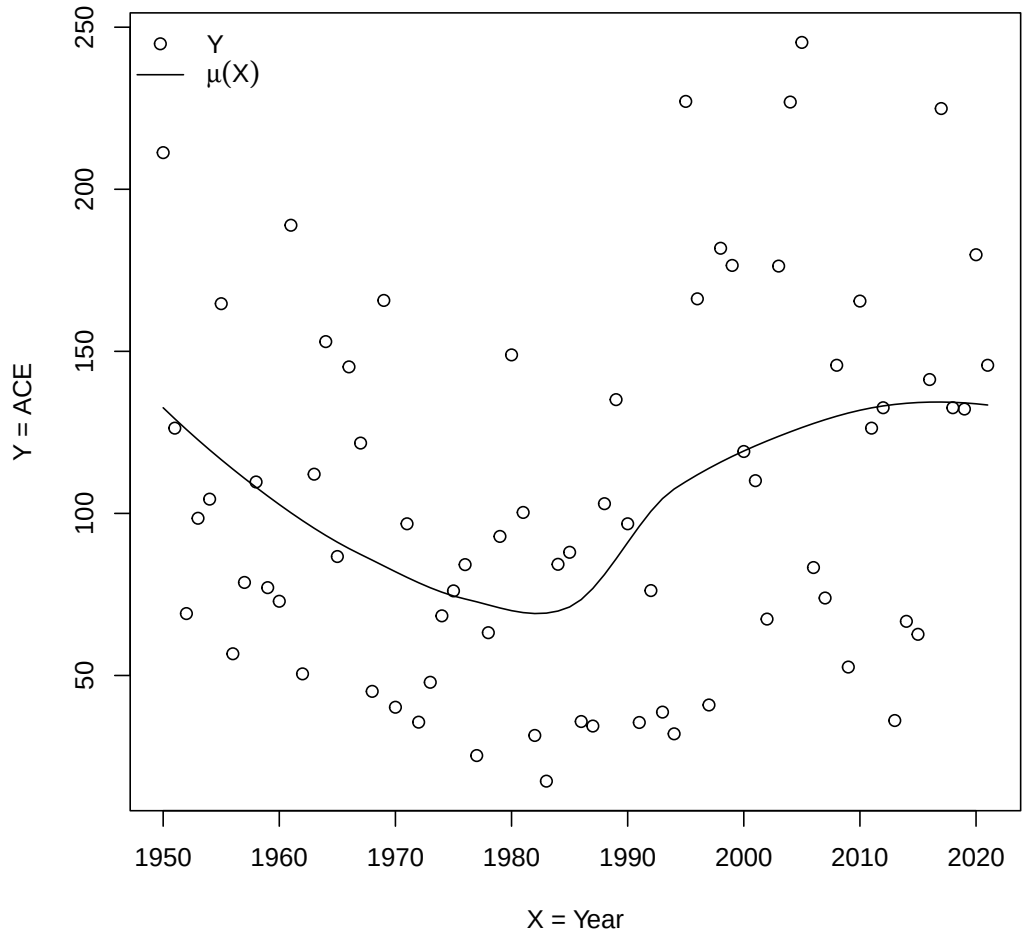
- Nonparametric regression attempts to estimate the regression function μ without strong assumptions

- The goal is to estimate a function μ rather than scalars β_j

Nonparametric regression

```
dat  <- read.csv(url("https://www4.stat.ncsu.edu/
        ~bjreich/ST740/hurricanes.csv"))
year <- dat[,1]
ACE  <- dat[year>1949,8]
year <- year[year>1949]
lo   <- loess(ACE~year)

plot(lo,xlab="X = Year",ylab="Y = ACE")
lines(lo$x,lo$fitted)
legend("topleft",c("Y",expression(mu(X))),
       pch=c(1,NA),lty=c(NA,1),bty="n")
```

Nonparametric regression



Accumulated Cyclone Energy (ACE) in the North Atlantic [4]

[4] http://tropical.atmos.colostate.edu/Realtime/index.php?arch&loc=northatlantic

Nonparametric regression

- We will specify priors on the function $\mu(\mathbf{X})$
- For example, say $p = 1$ and we use polynomial regression,

$$f(X) = \beta_0 + \sum_{j=1}^{m} X^j \beta_j$$

- This depends on m and parameters $\boldsymbol{\beta}_m = (\beta_0, ..., \beta_m)$
- The flexibility of the model is determined by its span
- For polynomial regression the span is the class of infinitely-differential functions, C
- Say the true regression function is $\mu_0 \in \mathcal{C}$, then there exists m and $\boldsymbol{\beta}_m$ so that $\mu(X) \approx \mu_0(X)$ for all X
- All models we will discuss span this (or similar) space [5]

[5] In deep learning this is called the "universal approximation theorem"

Nonparametric regression

- Bayesian NP regression uses many of the same models/ideas as classical NP regression
- The advantage of Bayesian methods are incorporation of prior information and uncertainty quantification
- Classical approaches often resort to plug-in estimators (e.g., the correlation parameters of a Gaussian process)
- In deep learning, Bayesian methods are the primary method for prediction intervals

Spline basis expansion

- A spline approximation (here $p = 1$) is

$$\mu(X) \approx \beta_0 + \sum_{j=1}^{m} B_j(X)\beta_j$$

  where $B_j(X)$ are fixed spline basis function and β_j are parameters to be estimated

- This expansion constructs m functions B_j to explain the effect of one variable, X

- There are many possibilities for B_j; we will use b-splines

- These are sparse piece-wise quadratic (by default) functions

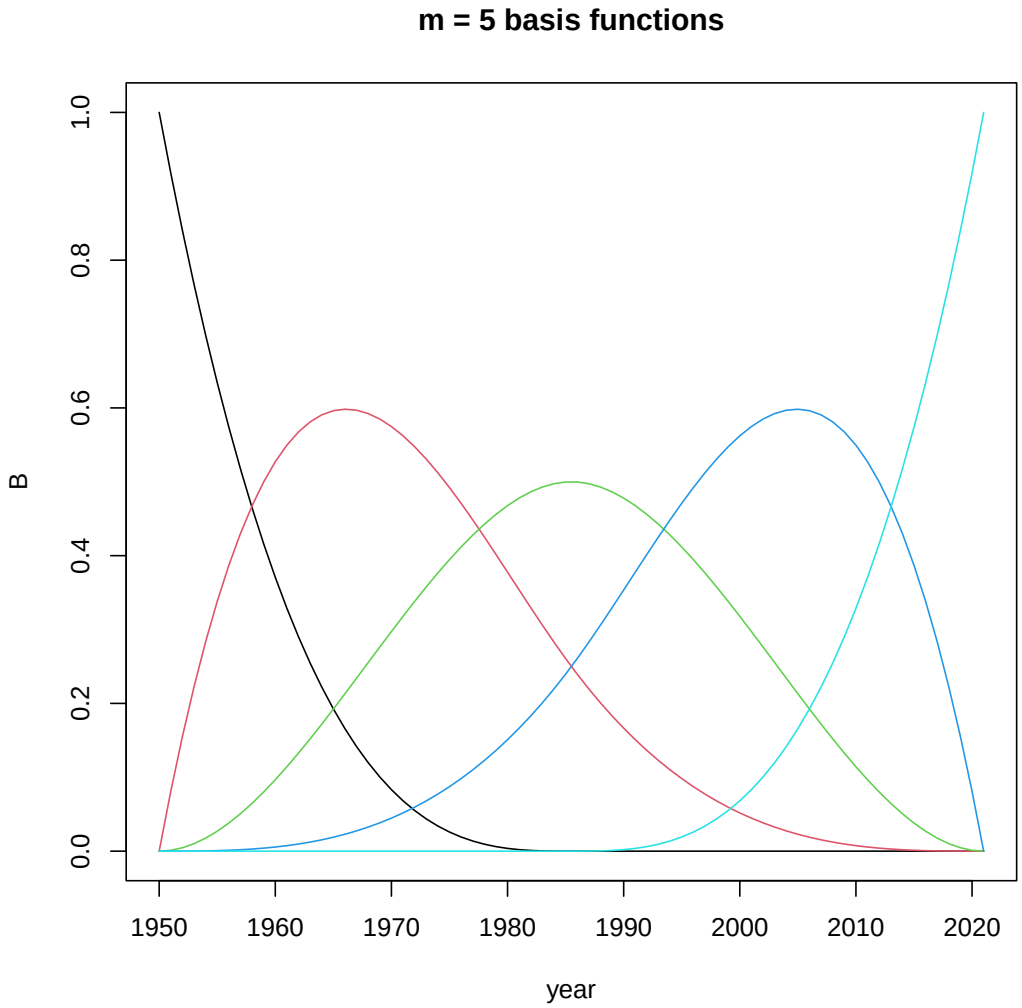
Bias-variance trade-off

- The β_j can be estimated by linear regression
- Large m can approximate any continuously differentiable function, but risks over-fitting
- Small m is more stable, but risks bias of the true μ is outside the span of the B_j
- Selecting m is critical

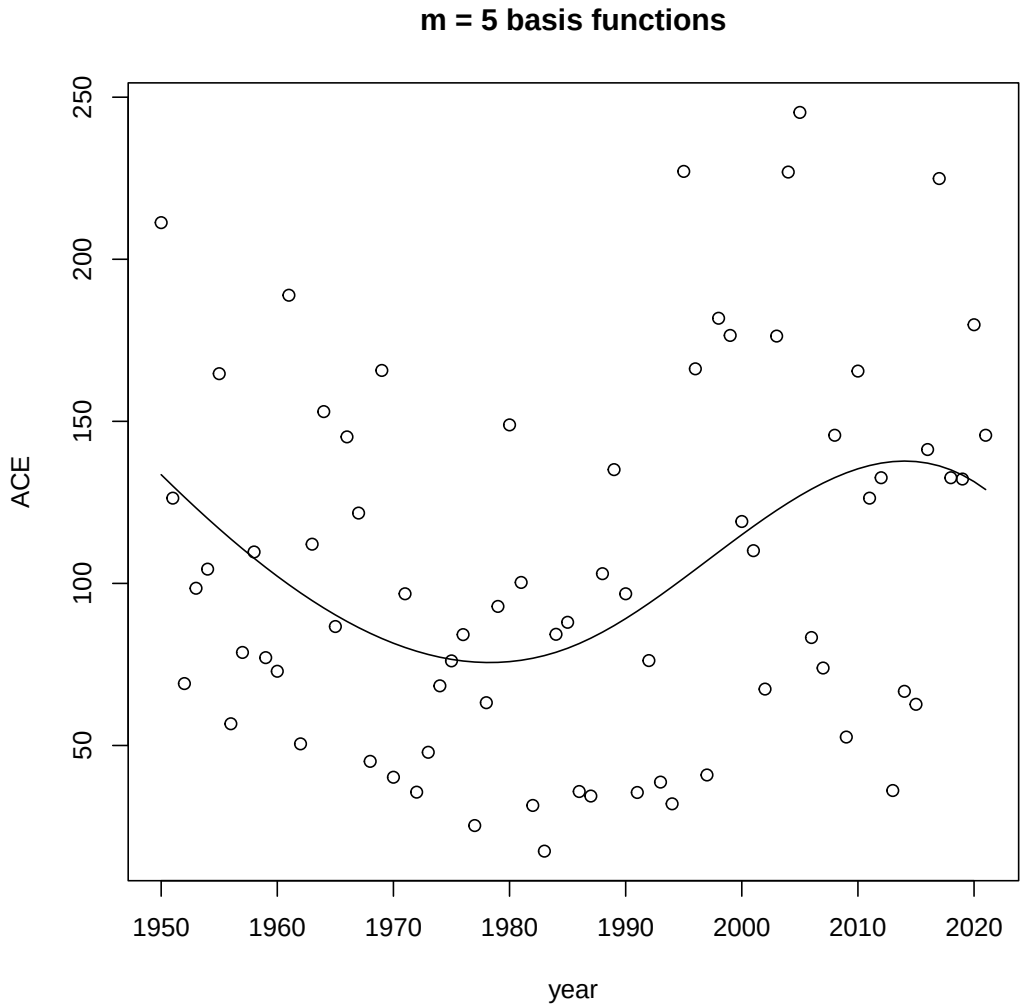
Nonparametric regression

```
library(splines)
m <- 5
B <- bs(year,df=m,intercept=TRUE)
dim(B)
[1] 75 5

matplot(year,B,type="l",lty=1,
        main=paste("m =",m,"basis functions"))

fit <- lm(ACE~B-1)

plot(year,ACE,main=
     paste("m =",m,"basis functions"))
lines(year,B%*%fit$coef)
```

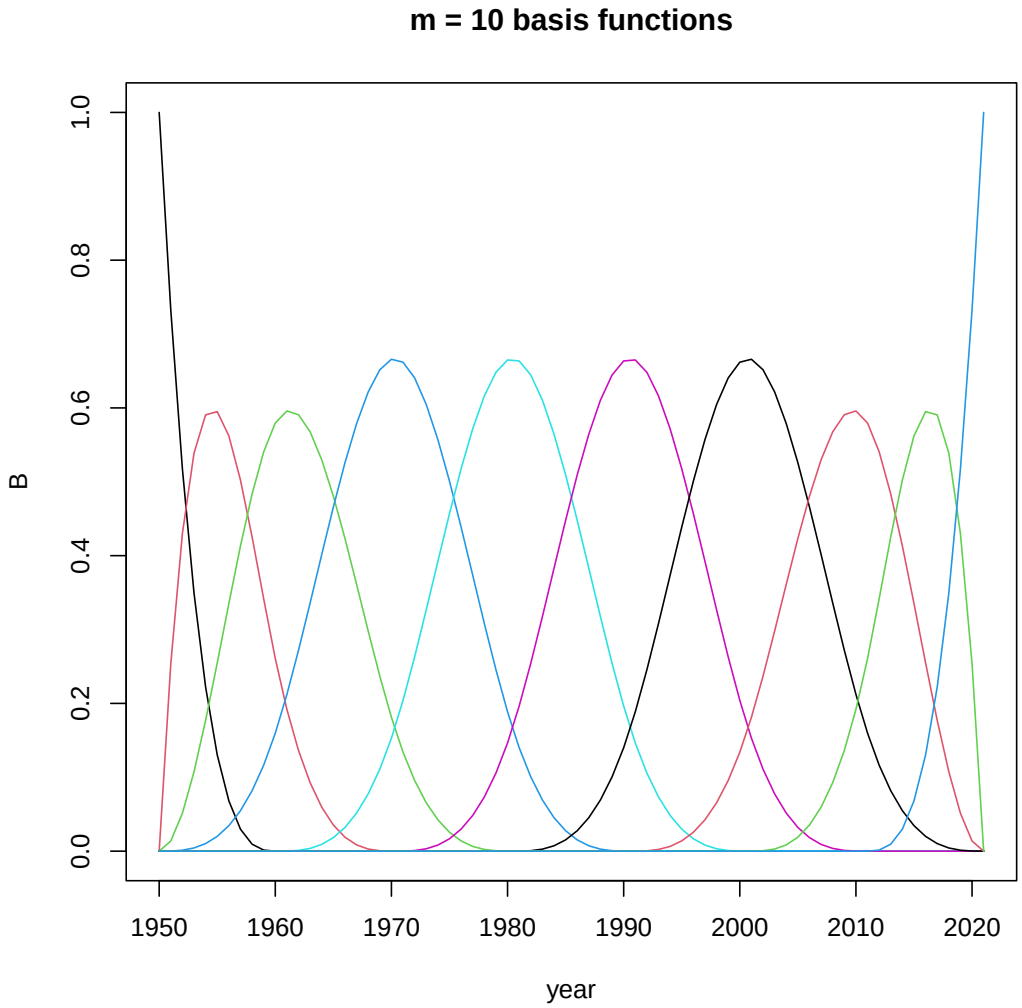
Spline regression



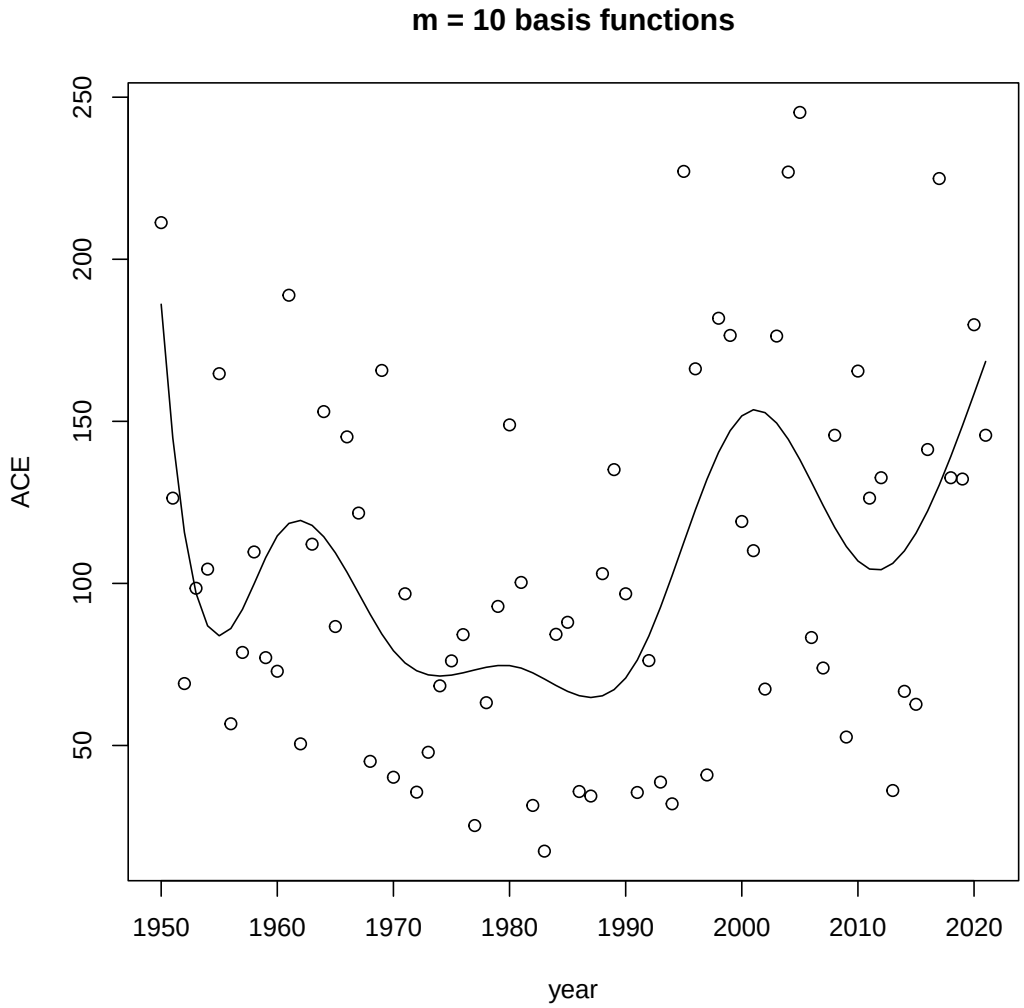
Spline regression



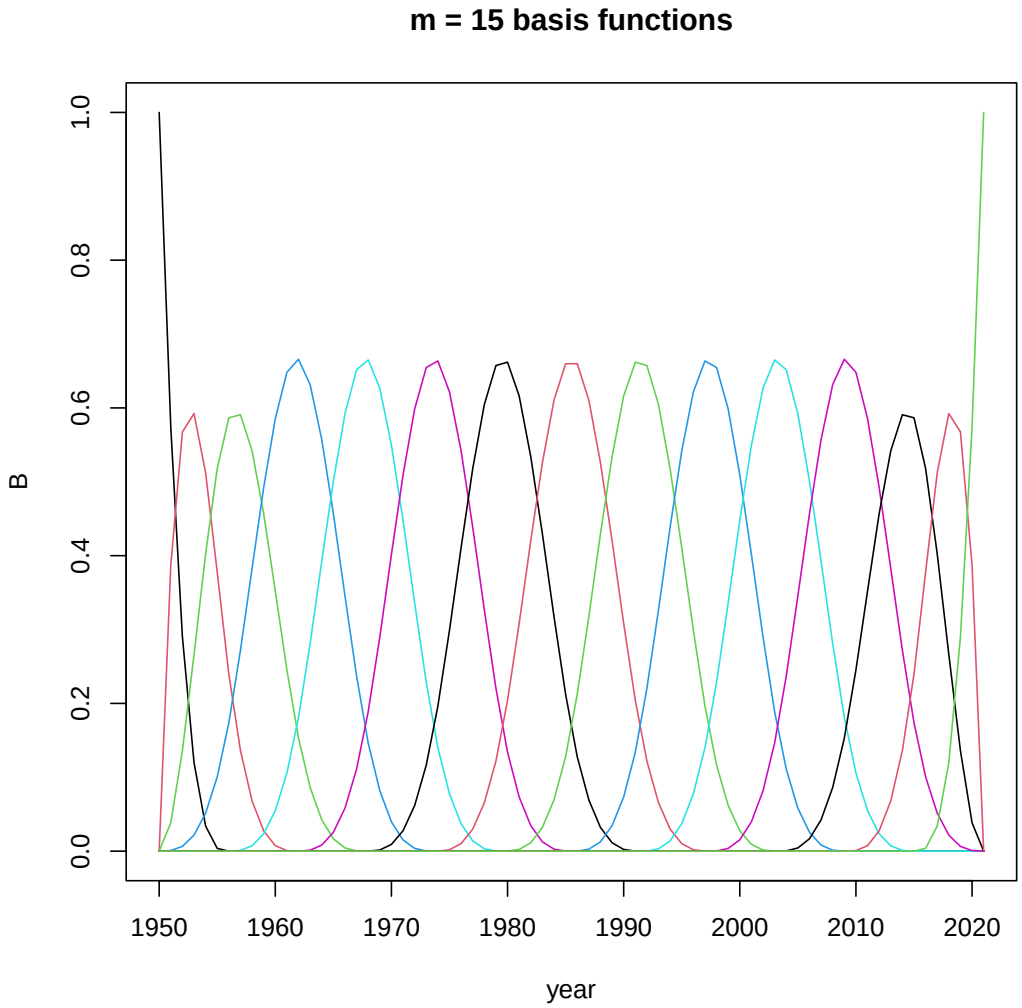
Spline regression



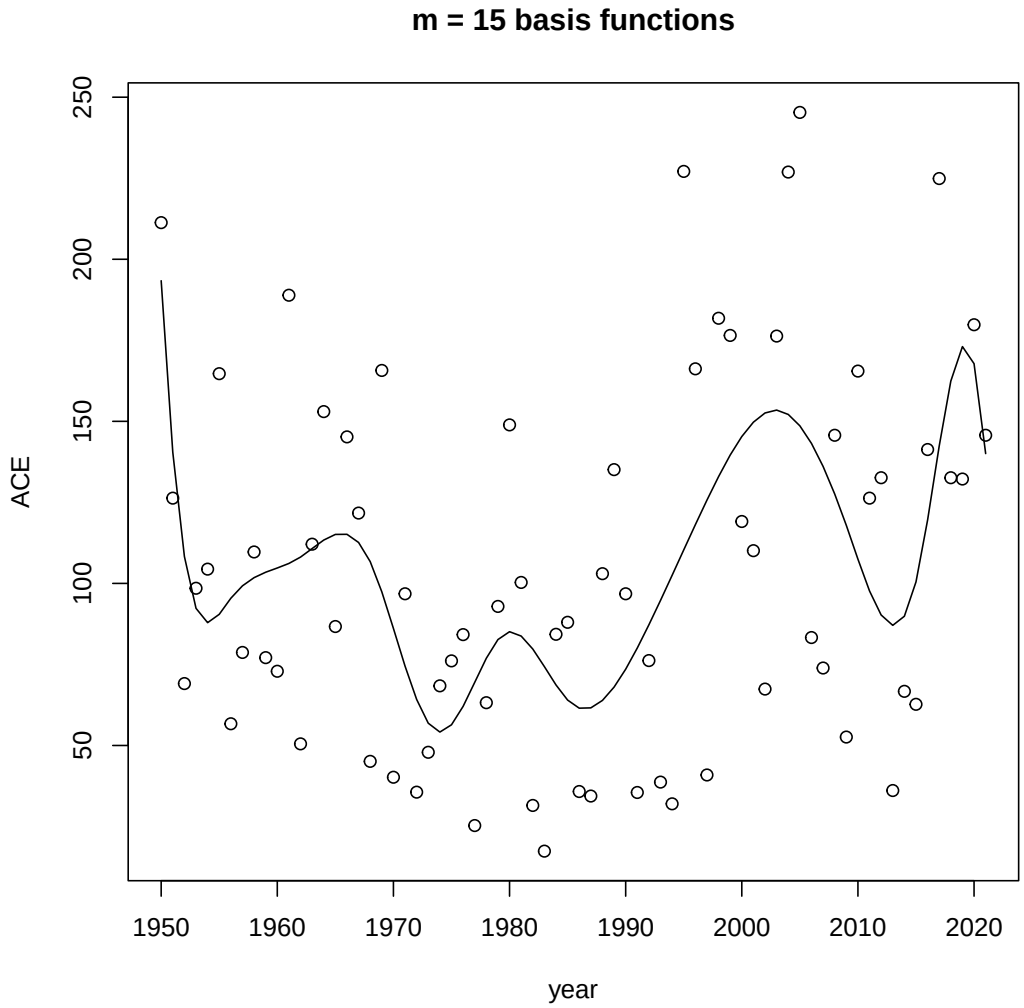
Spline regression



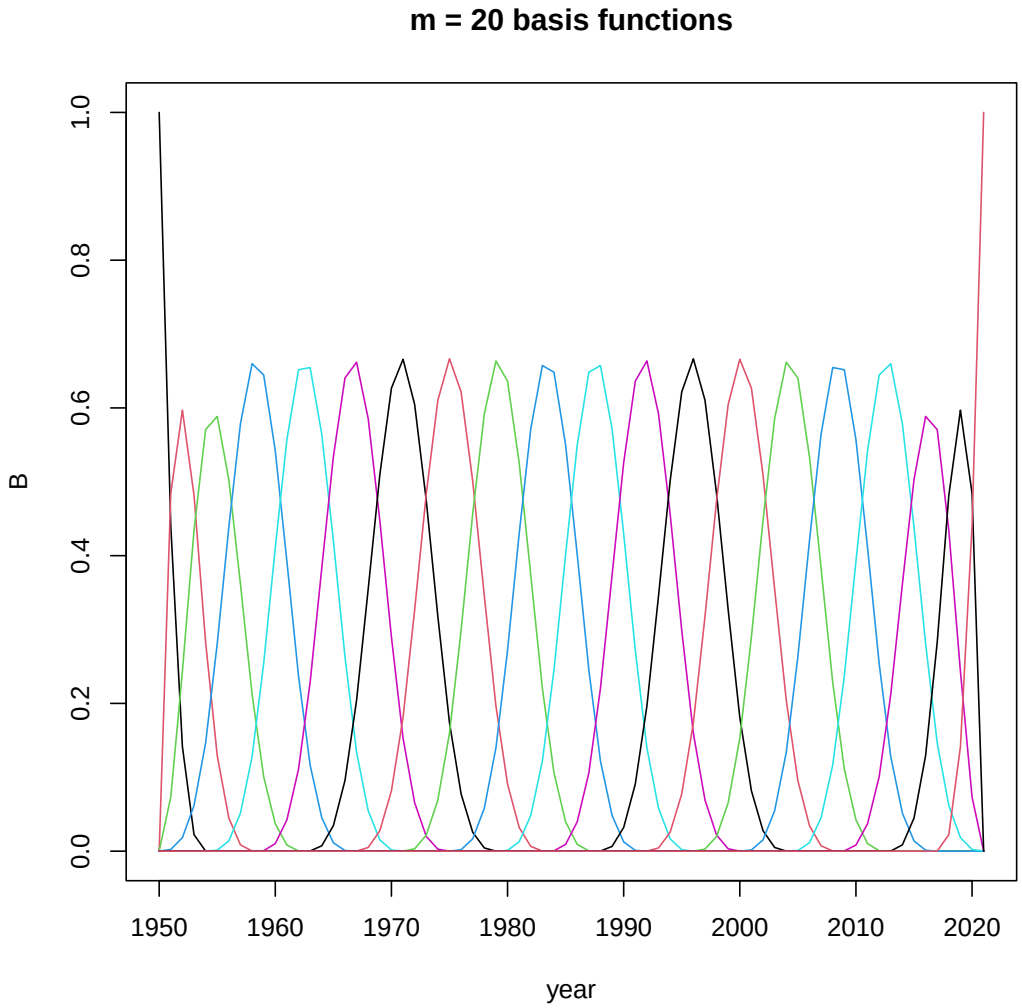
Spline regression



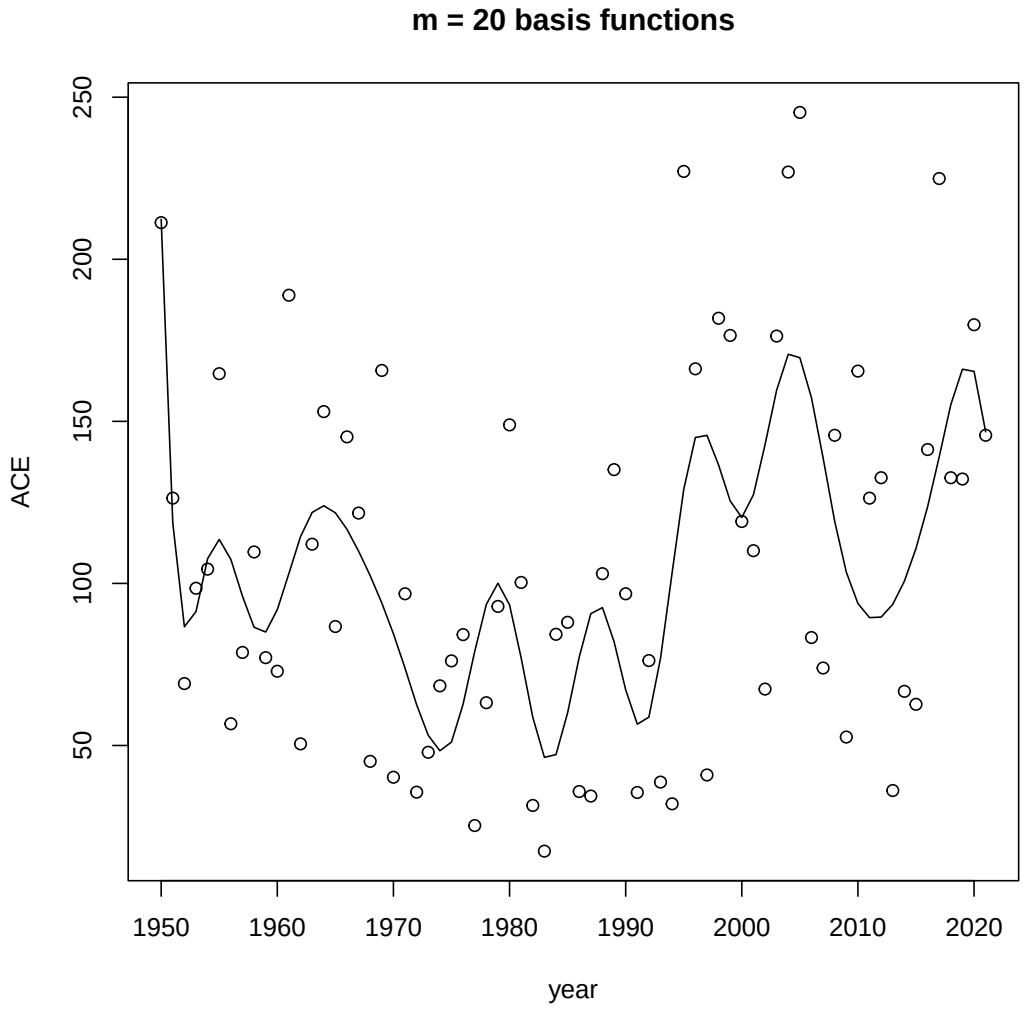
Spline regression



Spline regression



Spline regression

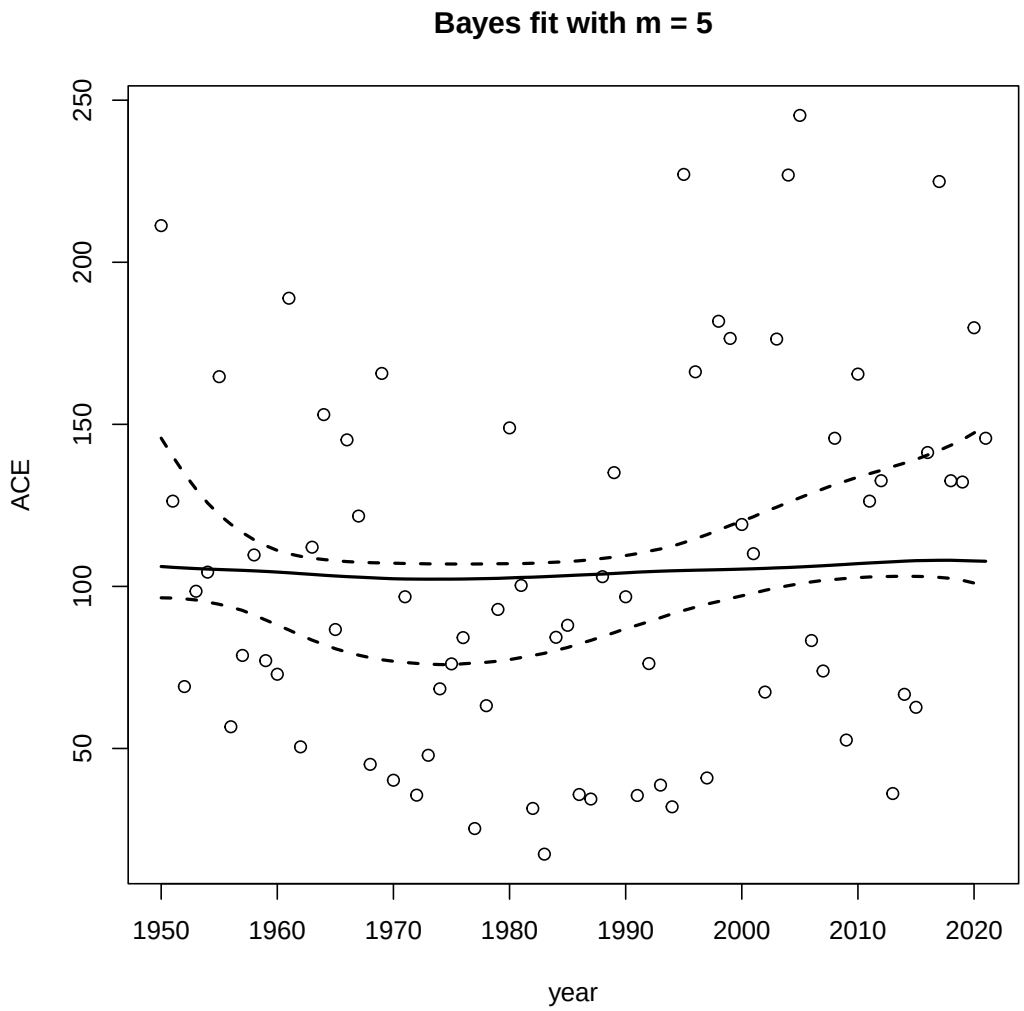


Bias-variance trade-off

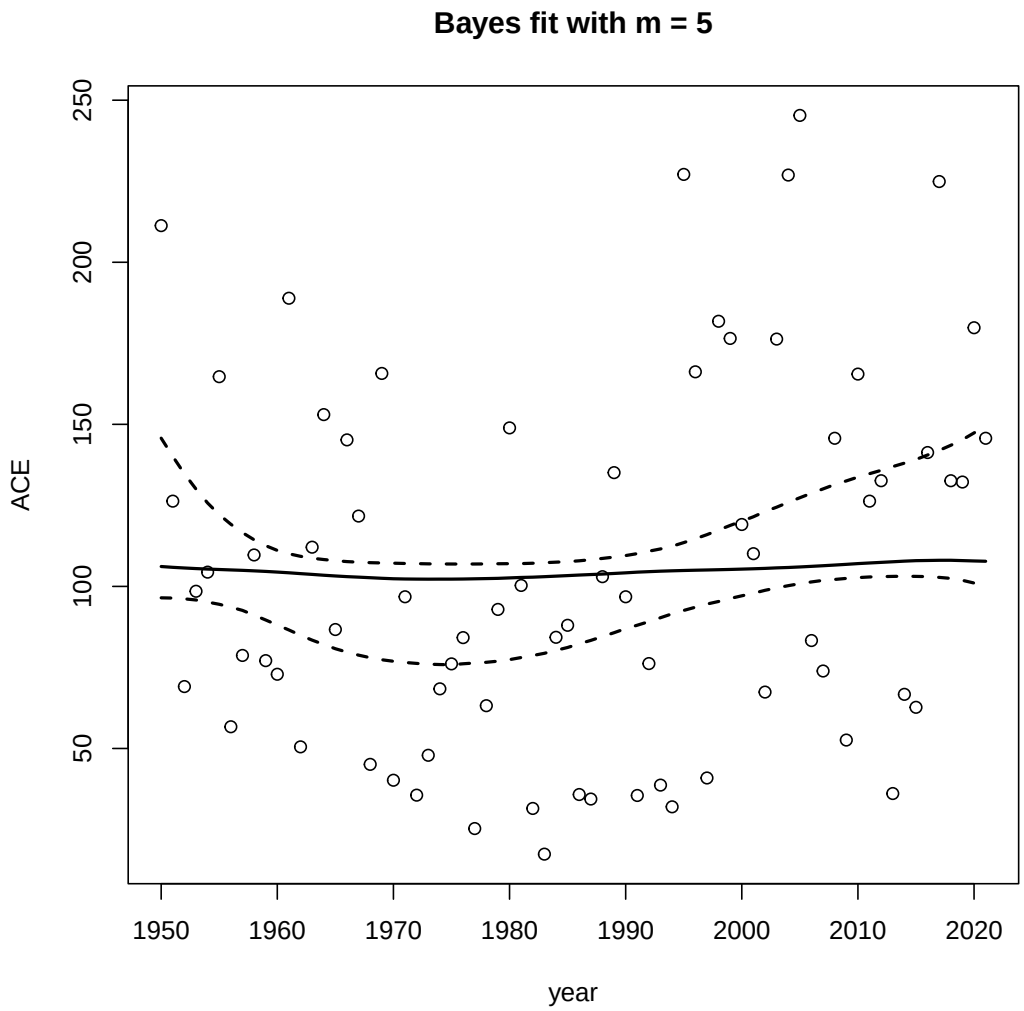
- The typical Bayesian approach[6] is to select m large enough to avoid bias, say $m = n$
- We then use priors to regulate the β_j and avoid overfitting
- The results on the next slides take $\beta_j|\tau \sim \text{Normal}(0, \tau^2)$ with $\tau \sim \text{InvG}$
- More sophisticated priors can be used, e.g., sparsity priors or priors with correlation across j

[6] Frequentists do similar things, often called "smoothing splines"

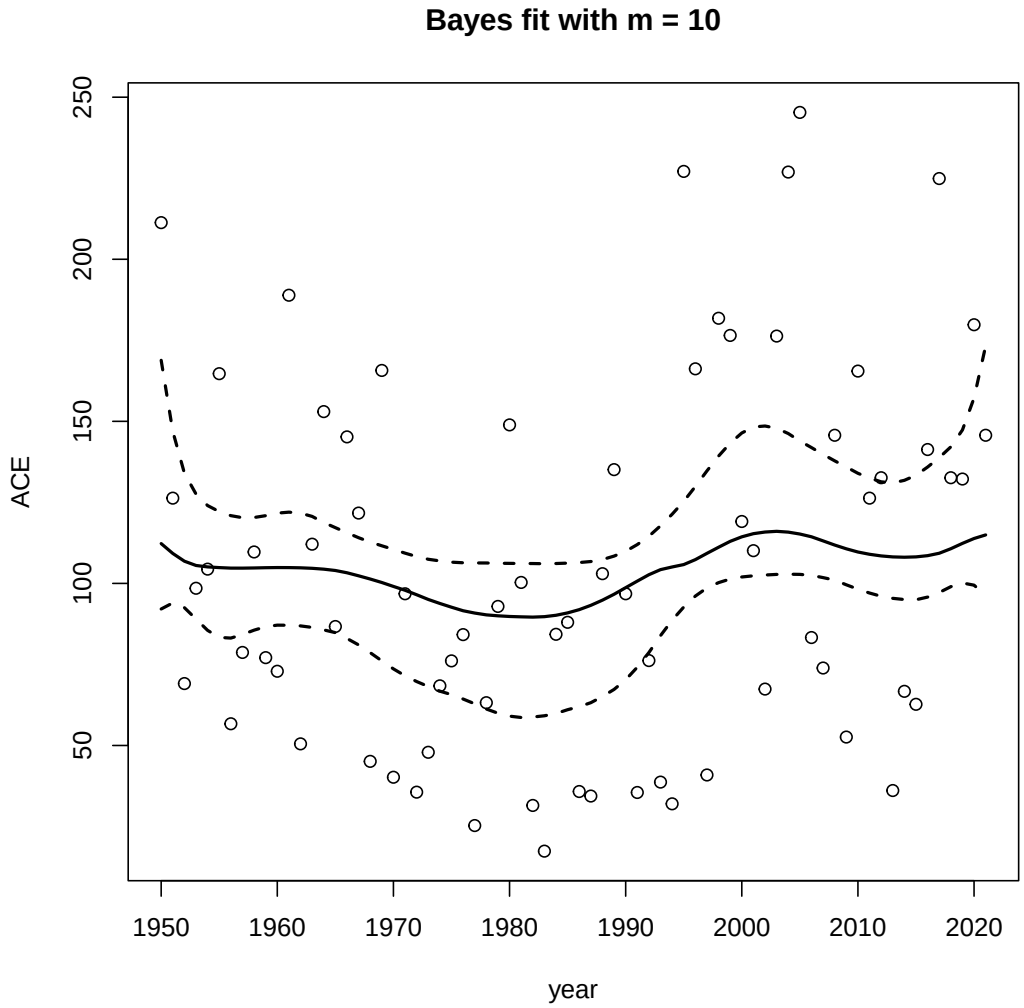
Bayesian spline regression



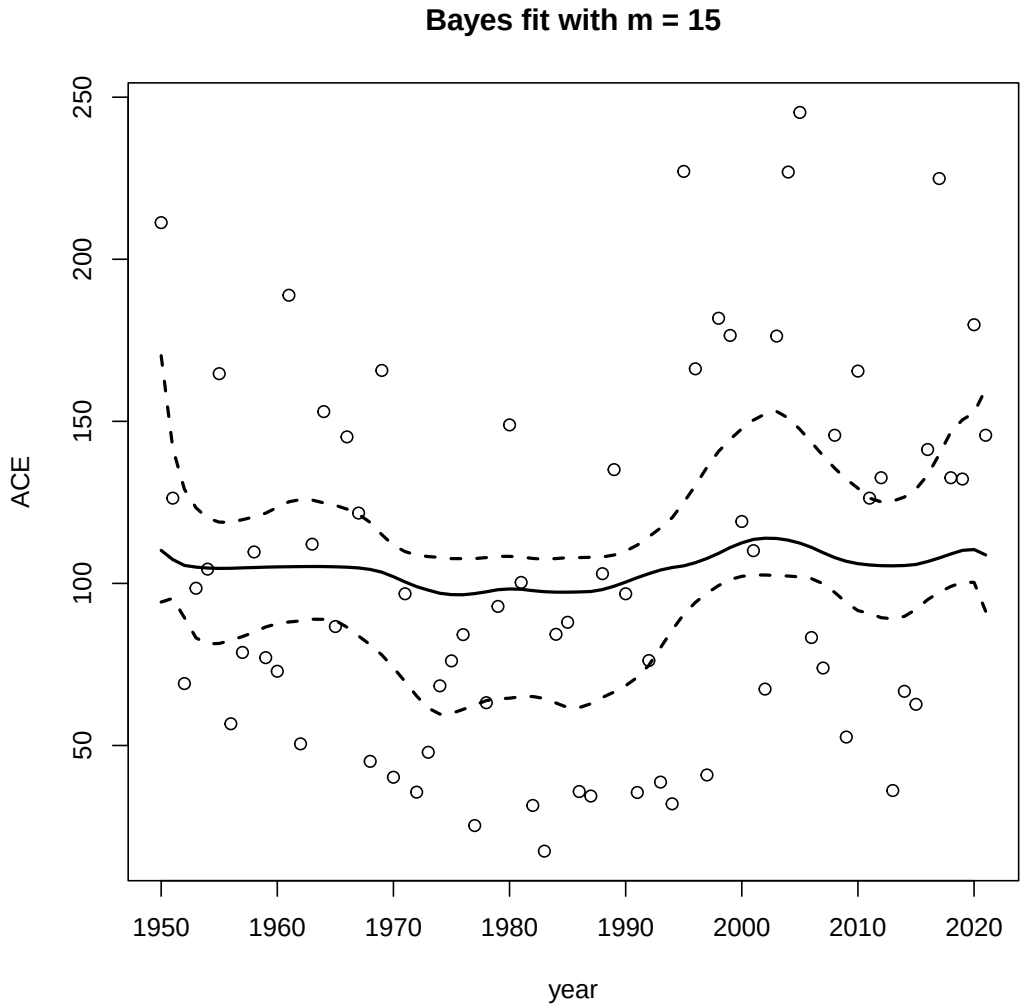
Bayesian spline regression



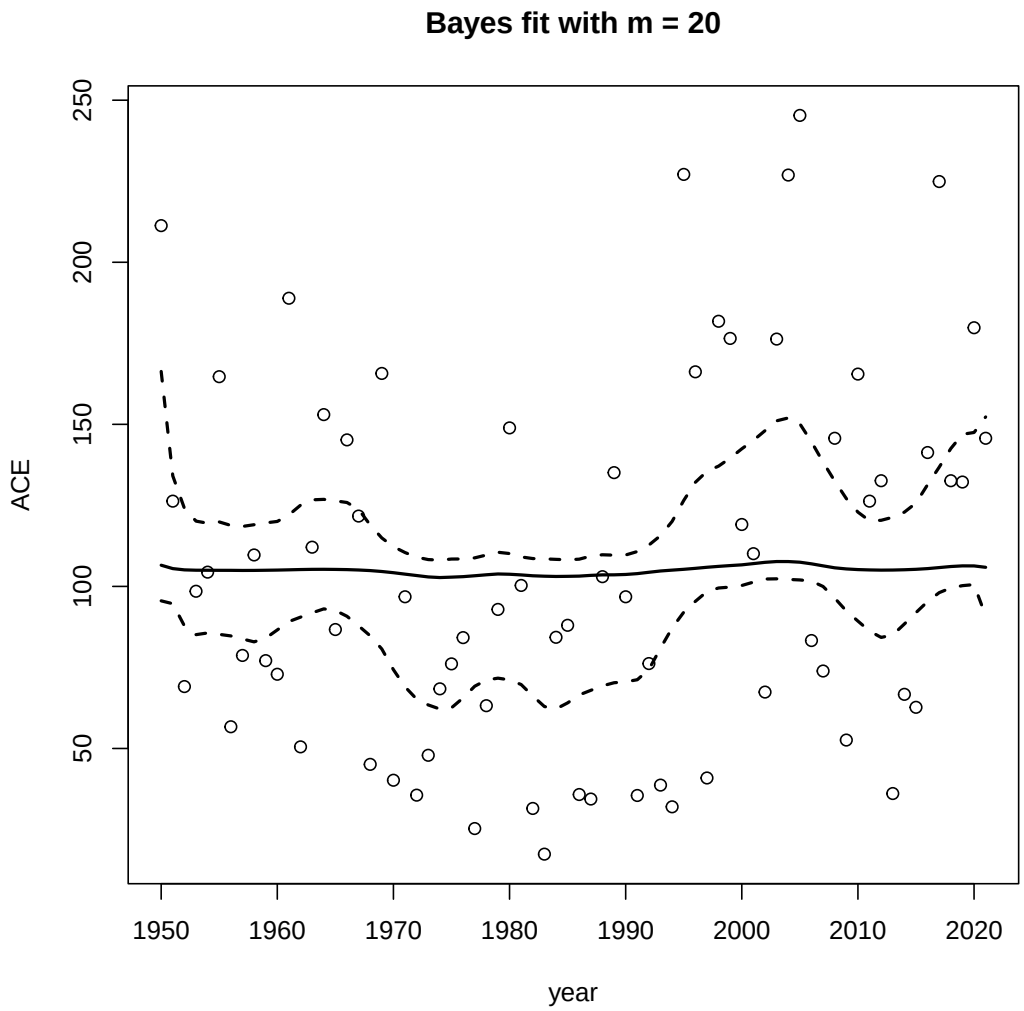
Bayesian spline regression



Bayesian spline regression



Bayesian spline regression



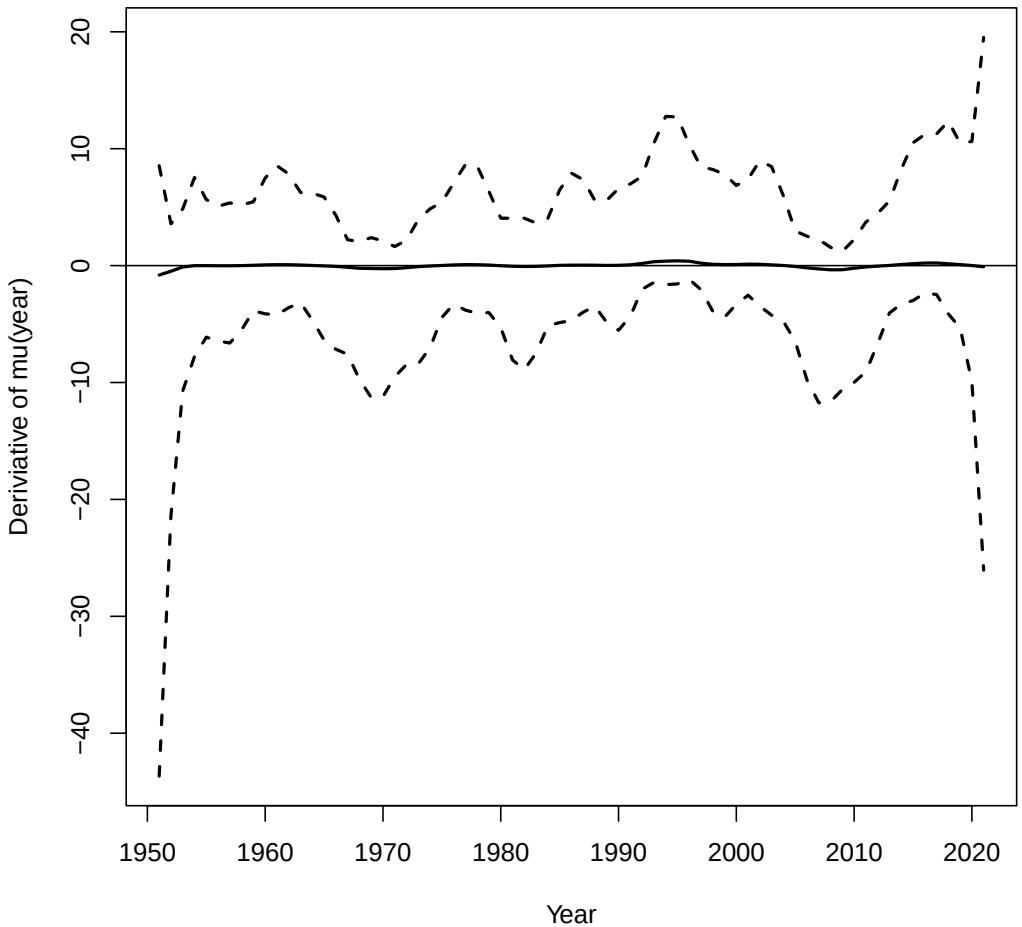
Bayesian spline regression

- If it is known that $\mu(X)$ is increasing in X, this can be incorporated in the prior
- For the b-splines, if $\beta_1 < ... < \beta_m$ then $\mu(X)$ is increasing
- You can also study the derivative

$$\frac{d\mu(X)}{dX} = \sum_{j=1}^{m} \frac{dB_j(X)}{dX} \beta_j$$

- It turns out that $dB_j(X)/dX$ is also a b-spline
- The next slide shows this function for the ACE data

Bayesian spline regression



Generalized additive models (GAMs)

- Extending spline regression for large p suffers from the curse of dimensionality

- If we desire a model for μ that can approximate any function on $\mathbf{X} \in \mathcal{R}^p$, we need m^p terms

$$\mu(\mathbf{X}) = \beta_0 + \sum_{j_1=1}^{m} \ldots \sum_{j_p=1}^{m} B_j(X_1) \cdot \ldots \cdot B_p(X_p)\beta_{j_1,\ldots,j_p}$$

- This has too many parameters for even moderate p

Generalized additive models (GAMs)

- GAMs reduce the dimension by assuming an additivity
- The main effects model is

$$\mu(\mathbf{X}) = \beta_0 + \sum_{j=1}^{p} f_j(X_j)$$

- Each of the p function has m terms,

$$f_j(X_j) = \sum_{l=1}^{m} B_{jl}(X_j)\beta_{jl}$$

- Prior might be $\beta_{jl} \sim \text{Normal}(0, \tau_j^2)$ with $\tau_j \sim \text{InvG}$
- This model has $pm << m^p$ terms and is interpretable

Generalized additive models (GAMs)

- A second-order model is

$$\mu(\mathbf{X}) = \beta_0 + \sum_{j=1}^{p} f_j(X_j) + \sum_{j<k}^{p} f_{jk}(X_j, X_k)$$

- The interaction terms are

$$f_{jk}(X_j, X_k) = \sum_{u=1}^{m}\sum_{v=1}^{m} B_u(X_j)B_v(X_k)\beta_{uvlk}$$

- This now has many parameters
- Wei et al [7] propose a Bayesian variable selection prior for additive models (priors with mass at $\tau_j = 0$)

[7] Wei, Reich, Hoppin, Ghoshal (2020). Sparse Bayesian additive nonparametric regression with application to health effects of pesticides mixtures. *Statistica Sinica*.

Outline

- High-dimensional data
  - Linear regression
  - Networks
- Nonparametric regression
  - Generalized additive models
  - **Bayesian additive regression trees**
  - Gaussian process regression
  - Bayesian deep learning
- Prior for a density function

Bayesian additive regression trees (BART)

- GAMs are efficient and interpretable, but struggle with high-order interactions
- In this sense they are not really nonparameteric because the can only fit a small class of regression functions
- Regression trees offer a simple way to handle high-order interactions
- Random forests are a classic way to fit tree models
- BART is a Bayesian alternative

Bayesian additive regression trees (BART)

- A tree model can also be written

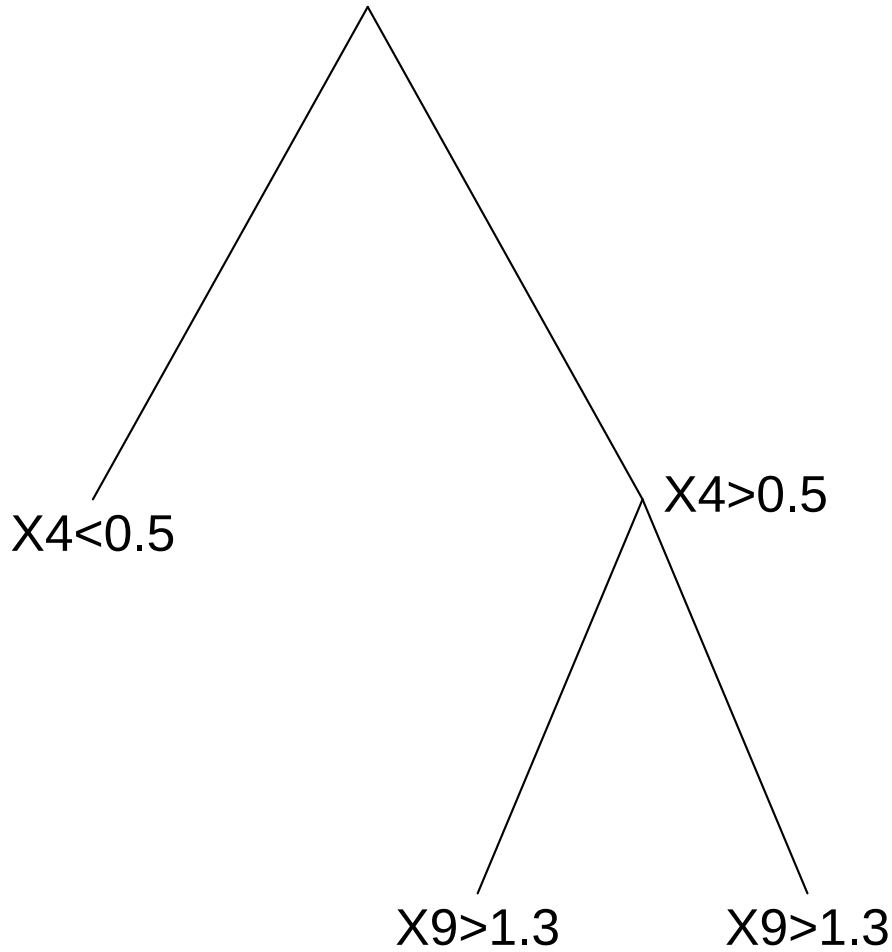
$$\mu(\mathbf{X}) = \sum_{l=1}^{m} B_l(\mathbf{X})\beta_l$$

  with $\beta_l \sim \text{Normal}(0, \tau^2)$

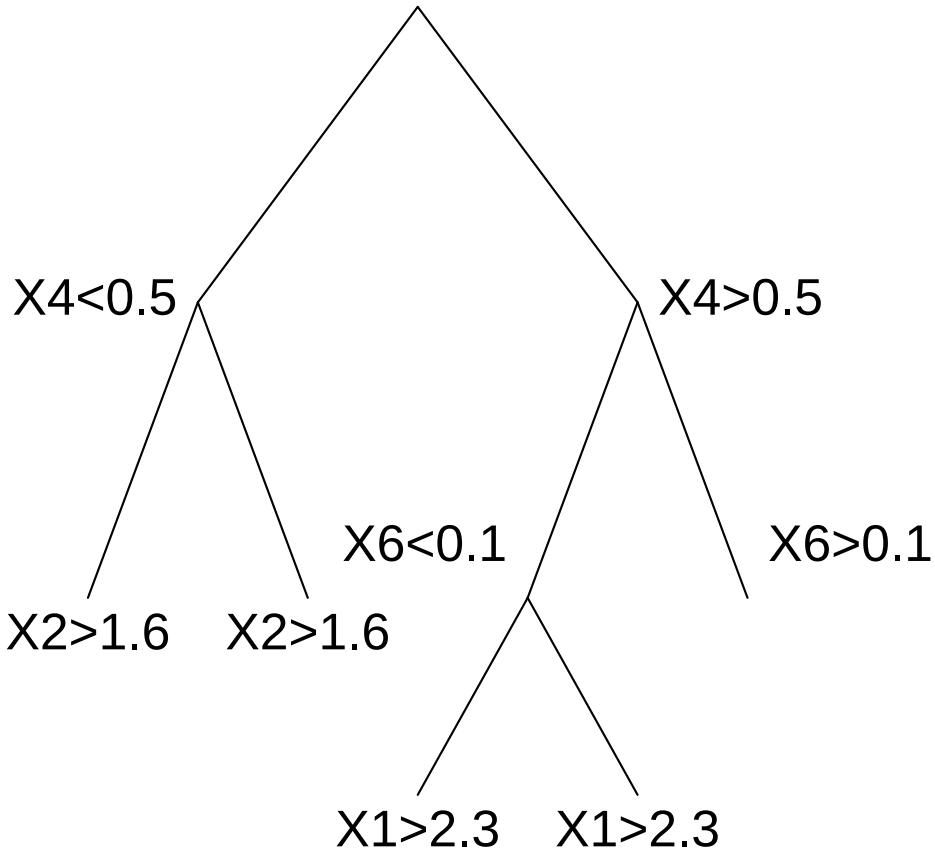
- However, the $B_l(\mathbf{X})$ are now leaves

- Example with $m = 3$:
  - $B_1(\mathbf{X}) = I(X_4 < 0.5)$
  - $B_2(\mathbf{X}) = I(X_4 > 0.5)I(X_9 < 1.3)$
  - $B_3(\mathbf{X}) = I(X_4 > 0.5)I(X_9 > 1.3)$

Small tree



Larger tree



Bayesian additive regression trees (BART)

- The variable (X_4) and threshold (0.5) in each branch ($X_4 < 0.5$) are unknown
- A Bayesian analysis puts priors on these, as well as the β_j
- BART averages over multiple trees

$$\mu(\mathbf{X}) = \sum_{k=1}^{K} \mu_k(\mathbf{X})$$

  where each μ_k is a tree with its own parameters

- This is challenging but implemented in the `R` package `BART`

Outline

- High-dimensional data
  - Linear regression
  - Networks
- Nonparametric regression
  - Generalized additive models
  - Bayesian additive regression trees
  - **Gaussian process regression**
  - Bayesian deep learning
- Prior for a density function

Gaussian process (GP) regression

- GP regression views $\mu(\mathbf{X})$ as a random function over $\mathbf{X} \in \mathcal{R}^p$
- The process μ is a GP if and only if all finite-dimensional distributions are MVN

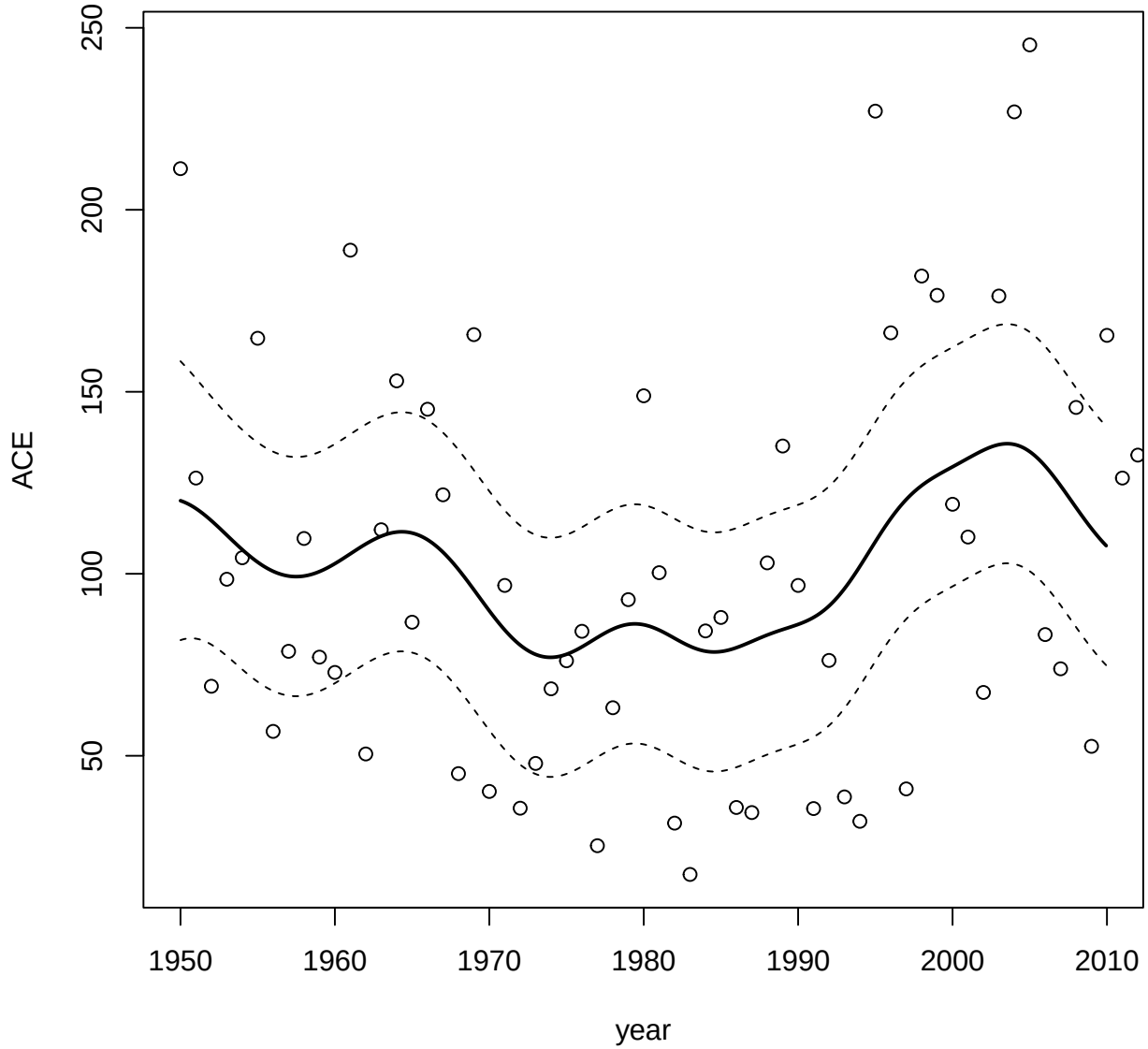
$$\mu_n = [\mu(\mathbf{X}_1), ..., \mu(\mathbf{X}_n)]^T \sim \text{Normal}(\mathbf{m}, \Sigma)$$

- A GP is defined by its mean and covariance functions
- Typically the mean function is constant $\text{E}[\mu(\mathbf{X}_i)] = m_i = \beta_0$
- The covariance function is often

$$\Sigma_{ij} = \text{Cov}[\mu(\mathbf{X}_i), \mu(\mathbf{X}_j)] = \tau^2 \exp[-(d_{ij}/\phi)^2]$$

for distance $d_{ij} = ||\mathbf{X}_i - \mathbf{X}_j||$

Gaussian process (GP) regression



Gaussian process (GP) regression

- The unknown parameters are $\boldsymbol{\theta} = \{\boldsymbol{\beta}, \sigma^2, \tau^2, \phi\}$
- The hierarchical model for $\mathbf{Y} = (Y_1, ..., Y_n)^T$ is

$$\mathbf{Y}|\boldsymbol{\mu}_n, \boldsymbol{\theta} \sim \text{Normal}(\boldsymbol{\mu}_n, \sigma^2\mathbf{I}_n) \quad \text{and} \quad \boldsymbol{\mu}_n|\boldsymbol{\theta} \sim \text{Normal}(\mathbf{m}, \Sigma)$$

- The model for $\mathbf{Y}$ marginal over $\boldsymbol{\mu}$, is

$$\mathbf{Y}|\boldsymbol{\theta} \sim \text{Normal}[\mathbf{m}(\boldsymbol{\theta}), \Sigma(\boldsymbol{\theta}) + \sigma^2\mathbf{I}_n]$$

- This is used to obtain the posterior of $\boldsymbol{\theta}$ via MCMC
- This is slow for large n

Gaussian process (GP) regression

- The predictive distribution of $\mu(X_{n+1})$ given $\mathbf{Y}$ is

$$\mu(X_{n+1})|\mathbf{Y}, \theta \sim \text{Normal}\left(m_{n+1} + P(\mathbf{Y} - \mathbf{m}), s^2\right)$$

- The mean operator is $P = \text{Cov}(\mu(X_{n+1}), \mathbf{Y})\Sigma^{-1}$

- The prediction variance is

$$s^2 = \text{Var}[\mu(X_{n+1})] - \text{Cov}[\mu(X_{n+1}), \mathbf{Y}]\Sigma^{-1}\text{Cov}[\mathbf{Y}, \mu(X_{n+1})]$$

- Both depend on θ, so samples of $\mu(X_{n+1})$ are drawn from the PPD using MCMC

- PPD samples for Y_{n+1} add σ^2 to s^2

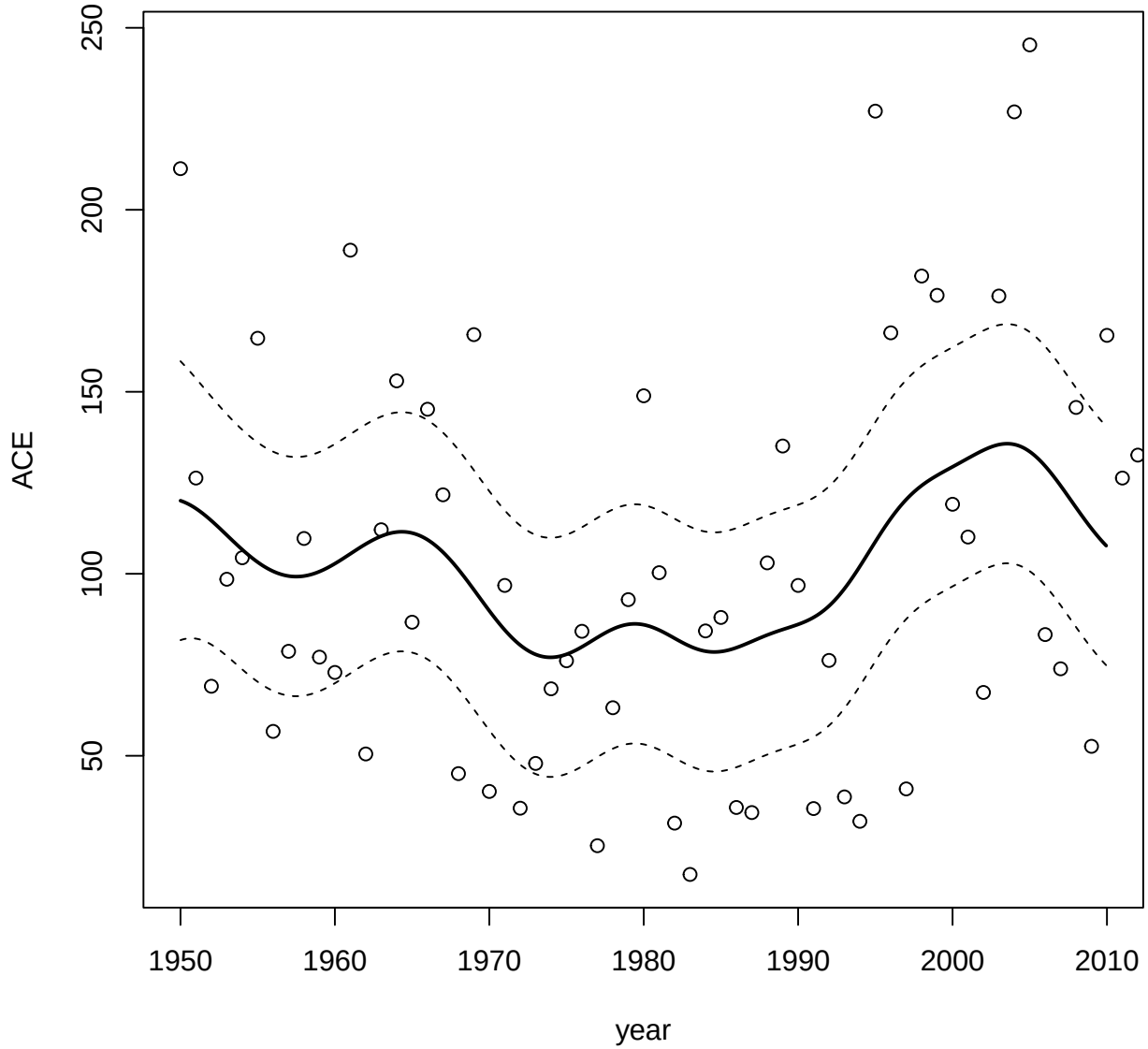
Gaussian process (GP) regression

```
yearp <- seq(1950,2010,0.1)
np    <- length(yearp)
sig2  <- 0.8*var(ACE) # Fixed for illustration
tau2  <- 0.2*var(ACE)
beta  <- mean(ACE)
phi   <- 5

d      <- as.matrix(dist(year))
SigInv <- solve(sig2*diag(n)+
                tau2*exp(-(d/phi)^2))
```

Gaussian process (GP) regression

```
m  <- v <- rep(0,np)

for(i in 1:np){
   dp   <- abs(yearp[i] - year)
   Sp   <- tau2*exp(-(dp/phi)^2)
   m[i] <- beta + Sp%*%SigInv%*%(ACE-beta)
   v[i] <- tau2 - t(Sp)%*%SigInv%*%Sp
}

plot(year,ACE,xlim=range(yearp))
lines(yearp,m,lwd=2)
lines(yearp,m-2*sqrt(v),lty=2)
lines(yearp,m+2*sqrt(v),lty=2)
```

Gaussian process (GP) regression



Gaussian process (GP) regression

- An anisotropic model allows variables to have different influence

$$\text{Cov}[\mu(\mathbf{X}_i), \mu(\mathbf{X}_k)] = \exp\left(-\sum_{j=1}^{p} \psi_j (X_{ij} - X_{kj})^2\right)$$

- If $\psi_j = 0$ covariate j is removed from the model

- A prior with mass at zero performs variable selection

Gaussian process (GP) regression

- In my opinion, GP is the gold standard for prediction for moderate p
- However, it is often very slow for even moderate n
- Computing $|\Sigma|$ and Σ^{-1} are bottlenecks
- Extending GPs to large n is an active area of research
- BART and deep learning are faster

Outline

- High-dimensional data
  - Linear regression
  - Networks
- Nonparametric regression
  - Generalized additive models
  - Bayesian additive regression trees
  - Gaussian process regression
  - **Bayesian deep learning**
- Prior for a density function

Deep learning

- We will discuss only a feed-forward neural network (FFNN)
- This assumes unstructured covariates like the other NP regression methods
- Deep learning is most powerful for structured covariates like images (CNN) or text (RNN)
- Deep learning architectures differ for these cases, but the Bayesian implementation is the same

Shallow learning

- FFNN starts with linear combinations (neurons) of the covariates (inputs)
- For neuron $l \in \{1, ..., L\}$, let

$$Z_l = b_l + \sum_{j=1}^{p} W_{jl} X_j$$

- This depends on the intercept (bias) b_l and slopes (weights) W_{jl}
- Non-linearity is introduced via the activation function ϕ, e.g., $\phi(x) = \text{expit}(x)$ or $\phi = x_+$
- In a GLM with link function g, the model is

$$g[\mathrm{E}(Y|\mathbf{X})] = \beta_0 + \sum_{l=1}^{L} \phi(Z_l)\beta_l$$

Deep learning

Deep learning adds K hidden layers

- Input layer: $Z_l^{(0)} = b_l^{(0)} + \sum_{j=1}^{p} W_{jl}^{(0)} X_j$ for $l \in \{1, ..., L_0\}$
- Hidden layer 1: $Z_l^{(1)} = b_l^{(1)} + \sum_{j=1}^{L_0} W_{jl}^{(1)} \phi_1(Z_l^{(0)})$ for $l \in \{1, ..., L_1\}$
- ...
- Hidden layer K: $Z_l^{(K)} = b_l^{(K)} + \sum_{j=1}^{L_{K-1}} W_{jl}^{(K)} \phi_K(Z_l^{(K-1)})$ for $l \in \{1, ..., L_K\}$
- Output layer: $g[\mathrm{E}(Y|\mathbf{X})] = b^{(K+1)} + \sum_{l=1}^{L_K} \phi_{K+1}(Z_l^{(K)}) W_l^{(K+1)}$

This spans $\mathcal{C}$ for large L, even with $K = 0$ hidden layers

Deep learning

- We need to estimate $\theta = \{b^{(0)}, ..., b^{(K+1)}, W^{(0)}, ..., W^{(K+1)}\}$
- A classical analysis selects θ to minimize an objective function, e.g., SSE or cross entropy
- As we've seen, this is equivalent to MAP estimation under a Gaussian or logistic regression model
- Classical analysis uses stochastic gradient descent, Bayesian uses MAP, HMC, SGMCMC or VB
- Classical analysis uses dropout to avoid overfitting, Bayesian uses (sparsity) priors

Variable selection

- SSVS/shrinkage priors can be used for variable selection
- If $W_{jl}^{(0)} = 0$ for all l then X_j is removed from the model
- SSVS prior: $W_{jl}^{(0)} = \delta_j w_{jl}^{(0)}$ for $\delta_j \sim$ Bernoulli and $w_{jl}^{(0)} \sim$ Normal
- Horseshoe prior: $W_{jl}^{(0)} \sim \text{Normal}(0, \delta_j^2 \sigma_0^2)$ with $\delta_j \sim$ HalfCauchy

Empirical Bayesian deep learning

- A frequentist analysis does not provide prediction uncertainty
- A full Bayesian analysis does, but it slow
- An empirical Bayesian analysis is a hybrid
- You first analyze the data using stochastic gradient descent and fix the parameters in the input and (some of) the hidden layers
- With these parameters fixed, you then conduct a shallow Bayesian analysis using MCMC

Deep Gaussian process approximation

- Another way to obtain prediction uncertainty is a GP approximation
- Consider even the shallow model
  - $Z_l = b_l + \sum_{j=1}^{p} W_{jl} X_j$
  - $g[\mathrm{E}(Y|\mathbf{X})] = \eta(\mathbf{X}) = \beta_0 + \sum_{l=1}^{L} \phi(Z_l)\beta_l$
- If the b_l, W_{jp} and β_l have normal priors, then $\eta(\mathbf{X})$ is approximately a GP for large L
- The covariance function is determined by ϕ and the prior variances

Outline

- High-dimensional data
  - Linear regression
  - Networks
- Nonparametric regression
  - Generalized additive models
  - Bayesian additive regression trees
  - Gaussian process regression
  - Bayesian deep learning
- **Prior for a density function**

Models for a density/distribution function

- We now have several flexible models a mean function
- A Bayesian model needs a full likelihood, not just the mean
- A nonparametric regression model is

$$Y_i = \mu(\mathbf{X}_i) + \varepsilon_i$$

  where the error distribution is $\varepsilon_i \sim f$ for distribution f
- A parametric model selects a family for f, say Gaussian
- A full NP Bayesian puts a prior on f
- Challenging because $f(e) \geq 0$ and $\int f(e)de = 1$

Semiparametric model

- A semiparametric model is a finite-mixture of normal
- The model is

$$f(e) = \sum_{j=1}^{m} q_j \phi(e; \gamma_j, \sigma^2)$$

  where ϕ is the Gaussian PDF
- Usually the probabilities have prior
  $q = (q_1, ..., q_m) \sim \text{Dirichlet}(\alpha_1, ..., \alpha_m)$
- The means γ_j can be fixed on a grid, or given prior
  $\gamma_j \sim \text{Normal}(0, \tau^2)$
- Increasing m can approximate any continuous PDF

Semiparametric model

- An alternative is a random histogram
- The model is

$$f(e) = \sum_{j=1}^{m} q_j U(e; b_j, b_{j+1})$$

  where U is the uniform PDF with fixed break points b_j

- The probabilities have prior
  $(q_1, ..., q_m) \sim \text{Dirichlet}(\alpha_1, ..., \alpha_m)$

Semiparametric model

- The hyperparameters α_j determine the prior
- Let $\alpha_j = cf_{0j}$ for $c > 0$ and $\sum_{j=1}^{m} f_{0j} = 1$
- Then prior mean is $\mathrm{E}(q_j) = f_{0j}$
- The prior variance is $\mathrm{V}(q_j) = f_{0j}(1 - f_{0j})/(c + 1)$
- Select f_{0j} can be based on a parametric model
- If the base distribution is standard normal then

$$f_{0j} = \int_{b_j}^{b_{j+1}} \phi(x)dx$$

- The concentration parameter c control prior strength

Dirichlet process prior (DPP)

- The DPP is a prior on a distribution function $F(e)$
- The base distribution is usually a parametric model, and can even have unknown parameters
- All draws from the prior are valid CDFs
- The prior support is $F \in \mathcal{C}$ where $\mathcal{C}$ is the collection all valid CDFs
- The DP has two hyperparameters: the base distribution $F_0(e)$ (a CDF) and concentration parameter $c > 0$

Dirichlet process prior (DPP)

- Like a GP, a DPP is defined by its finite-dimensional distributions
- Let $-\infty = b_1 < ... < b_{m+1} = \infty$ be an arbitrary set of breakpoints
- Define the probability in the intervals for the DPP as

$$P_j = F(b_{j+1}) - F(b_j)$$

- F follows a DPP if and only if

$$(P_1, ..., P_m) \sim \text{Dirichlet}(cf_{01}, ..., cf_{0m})$$

for $f_{0j} = F_0(b_{j+1}) - F_0(b_j)$

Dirichlet process prior (DPP)

- One way to draw approximate realizations is the stick-breaking representation
- The PMF corresponding to $F(e)$ can be written

$$f(e) = \sum_{j=1}^{\infty} p_j I(e = \gamma_j)$$

- The locations have prior $\gamma_j \sim f_0$
- The probabilities are $p_1 = v_1$ and for $j > 1$

$$p_j = v_j \prod_{k=1}^{j-1} (1 - v_k) = v_j \left(1 - \sum_{k=1}^{j-1} p_k \right)$$

and $v_j \sim \text{Beta}(1, c)$

Dirichlet process prior (DPP)

Derivation

Dirichlet process prior (DPP)

- For plotting and analysis, the infinite mixture can be truncated by setting $v_m = 1$ giving

$$f(e) = \sum_{j=1}^{m} p_j I(e = \gamma_j)$$

- The number of terms m set so that $\mathrm{E}(p_m)$ is small
- Another issue is that the DPP produces a discrete PMF
- A DP mixture of normals is continuous

$$f(e) = \sum_{j=1}^{m} p_j \phi(e; \gamma_j, \sigma^2)$$

where p_j and γ_i have priors as in the DPP

Other priors

- DPP can be generalized with different priors for the v_j, e.g., the Pitman-Yor process
- Finite mixtures model can be extended by having the number of terms, m, follow a Poisson prior
- The Polya tree prior is a tree-based prior for a PDF
- Density regression allows the mixture locations and/or probabilities to depend on covariates,

$$f(e; x) = \sum_j p_j(x)\phi\left(e; \lambda_j(x), \sigma^2\right)$$