Module 1: Nonparametric Preliminaries

Selecting Smoothing Parameters

STAT/BIOSTAT 527, University of Washington
Emily Fox
April 8th, 2014

Smoothing Parameter

- In both ridge and lasso regression, we saw that the parameter λ controlled the solution
  - Often, can straightforwardly equate with effective degrees of freedom
- Which λ (→ estimator) should we choose???

Two Goals

- **Model Selection:** estimating the performance of models in order to select the best one
  - E.g., choosing λ

- **Model Assessment:** having chosen a final model, estimate its prediction error (generalization error) on new data

- Ideally, divide data into 3 parts

TRAIN	VALIDATION	TEST

Focus on Model Selection

- Which estimator/smoothing parameter should we choose?

TRAIN	VALIDATION

- Recall metrics for assessing the performance of an estimator…

Measuring Predictive Performance

- Assume estimate $\hat{f}_n(\cdot)$ based on training data $y_1, \ldots, y_n$
- The ***generalization error*** provides a measure of predictive performance

$$GE(\hat{f}_n) = E_{Y,X}\left[L(Y, \hat{f}_n(X))\right]$$

Measuring Predictive Performance

- Assume L_2 loss $Y = f(x) + \epsilon$ $E[\epsilon] = 0$ $\mathrm{var}(\epsilon) = \sigma^2$
- Averaging over repeat training sets $\mathbf{Y}_n = Y_1, \ldots, Y_n$ we get the ***predictive risk*** at x^*

$$E_{Y^*,\mathbf{Y}_n}\left[(Y^* - \hat{f}_n(x^*))^2\right] = E_{Y^*,Y_n}\left[(Y^* - f(x^*) + f(x^*) - \hat{f}_n(x^*))^2\right]$$

test   training   fcn of training data Y_n

$$= E_{Y^*}\left[(Y^* - f(x^*))^2\right] + \underbrace{E_{Y_n}\left[(\hat{f}_n(x^*) - f(x^*))^2\right]} + 2\cancel{E_{Y^*}\left[Y^* - f(x^*)\right]}\,E_{Y_n}\left[\hat{f}_n(x) - f(x^*)\right]$$

$\to 0$

$$= \sigma^2 + MSE(\hat{f}_n(x^*))$$

"irreducible error"   "risk"

- Recall $MSE[\hat{f}_n(x)] = \mathrm{bias}(\hat{f}_n(x))^2 + \mathrm{var}(\hat{f}_n(x))$

Measuring Predictive Performance

- Finally, let's average over covariates *x*

(focus on MSE bc can't avoid σ^2)

  - ***Integrated MSE***

$$\int MSE(\hat{f}_n(x))\,p(x)\,dx$$

summary over all inputs

  - ***Average MSE***

$$\frac{1}{n}\sum_{i=1}^{n} MSE(\hat{f}_n(x_i))$$

Monte Carlo est: $x_i \sim P \quad i=1,\ldots,n$

- Note: ***avg. pred. risk =*** σ^2 + ***avg. MSE***

still our focus

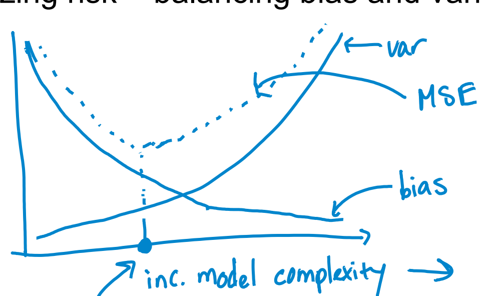
$$\left[\frac{1}{n}\sum_{i=1}^{n} E_{Y_i^*, Y_n}\left[(Y_i^* - \hat{f}_n(x_i))^2\right]\right]$$

Y_i^*: new obs. at x_i; Y_n: training data

Bias-Variance Tradeoff

recall polynomial reg. example

- Minimizing risk = balancing bias and variance



optimal solution for model selection task

MSE is a fcn of $f(\cdot)$

- Note: *f(x) is unknown, so cannot actually compute MSE*

Focus on Model Selection

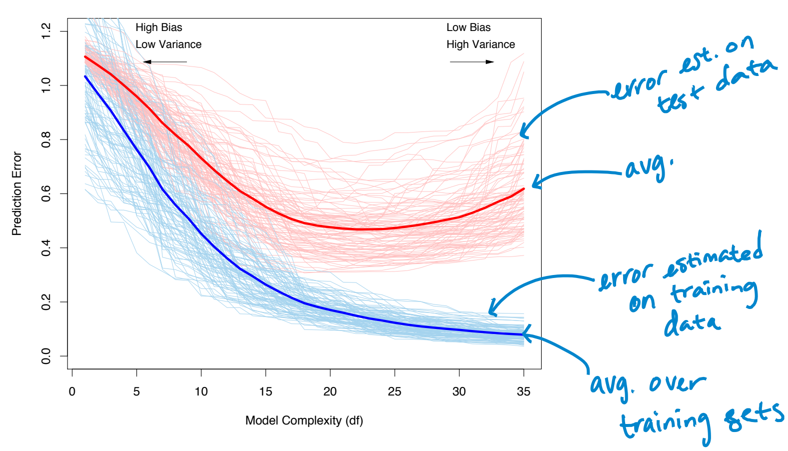
- Which estimator/smoothing parameter should we choose?

TRAIN	VALIDATION

- We saw that minimizing (average) prediction error can be equated with minimizing (average) MSE
- With a validation set, we can estimate the prediction error

In Practice…

- Minimizing risk = balancing bias and variance



From Hastie, Tibshirani, Friedman

Data Scarce Approximations

- Often, we do not have enough data to form suitably sized training and validation sets
  - What is a good training/test split? Sensitivity?
  - Typically want to use as much data for training as possible

- Rely on other approximations

Approx 1: Training Data Only

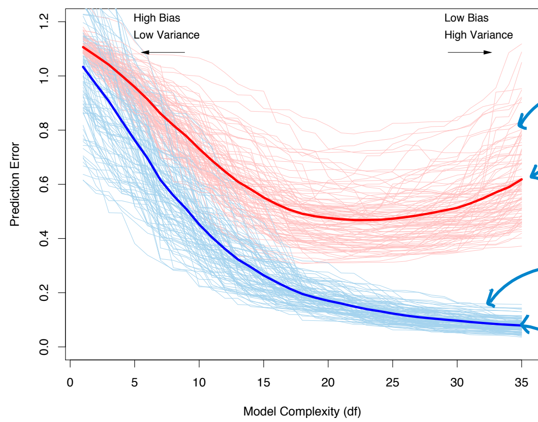
- **Goal:** Minimize average MSE

$$\min_{\lambda} E\left[\frac{1}{n}\sum_{i=1}^{n}(f(x_i) - \hat{f}_n^{\lambda}(x_i))^2\right]$$

- **Solution:** Use training error

In Practice…

- Minimizing risk = balancing bias and variance



error est. on test data

avg.

error estimated on training data

avg. over training sets

From Hastie, Tibshirani, Friedman

Approx 2: Cross Validation

- **Goal:** Minimize average MSE

$$\min_{\lambda} E\left[\frac{1}{n}\sum_{i=1}^{n}(f(x_i) - \hat{f}_n^{\lambda}(x_i))^2\right]$$

- **Solution:** Mimic heldout data using *training* data

- Leave-one-out (LOO) cross validation (CV) algorithm:
  - Estimate fit using all but i^{th} data point
  - Predict i^{th} observation
  - Repeat for all i

  - Repeat for all values of λ

Approx 2: Cross Validation

- Reasoning

- For linear smoothers

- Warning: Curves can be very flat…Don't just choose and use without thinking. Some rules of thumb (see Elements of Statistical Learning)

Approx 2: Cross Validation

- K-fold cross validation

TRAIN	TRAIN	VALID-ATION	TRAIN	TRAIN

- Algorithm
  1. Fit model using data with k^{th} fraction removed
  2. Using fitted model, compute

$$\mathrm{CV}_k = \frac{1}{n_k} \sum_{i \in J(k)} (y_i - \hat{f}^{\lambda}_{-k}(x_i))$$

  3. Store

$$\mathrm{CV} = \frac{1}{K} \sum_{k=1}^{K} \mathrm{CV}_k$$

  4. Repeat for each value of λ using same split of the data

Approx 3: Generalized CV

- Recall LOO ordinary CV for linear smoothers

$$CV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{f}_n^{\lambda}(x_i)}{1 - L_{ii}}\right)^2$$

- Instead of L_{ii}, use $\frac{1}{n}\sum_{i=1}^{n} L_{ii}$

- Often very close to OCV solution

Approx 3: Generalized CV

$$GCV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{f}_n^{\lambda}(x_i)}{1 - \frac{\nu_\lambda}{n}}\right)^2$$

- One motivation: Invariance to orthonormal transformations

Approx 3: Generalized CV

$$GCV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{f}_n^{\lambda}(x_i)}{1 - \frac{\nu_\lambda}{n}}\right)^2$$

- Using $(1-x)^{-2} \approx 1 + 2x$

Approx 4: Mallows C_p Statistic

- **Goal:** Minimize average MSE

$$\min_{\lambda} E\left[\frac{1}{n}\sum_{i=1}^{n}(f(x_i) - \hat{f}_n^{\lambda}(x_i))^2\right]$$

- **Solution:** Approximate directly

$$\text{avg. MSE} = \frac{1}{n}E\left[(f - \hat{f}_n^{\lambda})^T(f - \hat{f}_n^{\lambda})\right]$$

Approx 4: Mallows C_p Statistic

$$\text{avg. MSE} = \frac{1}{n} E\left[(Y - L^\lambda Y)^T (Y - L^\lambda Y)\right] - \sigma^2 + \frac{2}{n}\nu_\lambda \sigma^2$$

- Estimate avg. MSE as

- Note: Arises from considering L_2 loss. Log-likelihood loss leads to AIC. For BIC, consider Bayesian model selection

Bayesian Model Selection

- Assume some *M* possible models
  - Model M_m *m=1,…,M* has parameters θ_m and prior $p(\theta_m \mid M_m)$
  - Prior over models $p(M_m)$

- Model posterior

$$p(M_m \mid Z) \propto p(M_m) p(Z \mid M_m)$$
$$\propto p(M_m) \int p(Z \mid \theta_m, M_m) p(\theta_m \mid M_m) d\theta_m$$

- Compare models:

$$\frac{p(M_m \mid Z)}{p(M_\ell \mid Z)} = \frac{p(M_m) p(Z \mid M_m)}{p(M_\ell) p(Z \mid M_\ell)} \gtrless 1$$

Bayesian Model Selection

- For Bayes factor, approximate
$$\log p(Z \mid M_m) \approx \log p(Z \mid \hat{\theta}_m, M_m) - \frac{\nu_m}{2} \log n + O(1)$$
- If loss is $-2 \log p(Z \mid \hat{\theta}_m, M_m)$, then equivalent to BIC
  - Minimizing BIC = maximizing approximated posterior
- However, in addition to being able to select the best model, in Bayesian framework we also get the relative merit of each
$$\approx \frac{e^{-\frac{1}{2}\mathrm{BIC}_m}}{\sum_{\ell=1}^{M} e^{-\frac{1}{2}\mathrm{BIC}_\ell}}$$
- BIC is asymptotically consistent, but AIC is not
- For finite samples, BIC tends to choose too simple models

Reading

- Hastie, Tibshirani, Friedman: 7.2 (again), 7.4-7.7, 7.10
- Wakefield: 10.6 (up to 10.6.4)

What you should know…

- Model selection vs. model assessment tasks
- Training/validation/test split
- In-sample approaches for selecting the smoothing parameters:
  - Training error = BAD
  - Cross validation (CV)
    - LOO
    - K-fold
  - Generalized cross validation (GCV)
  - Mallow's C_p
- Bayesian model selection

Module 2: Splines and Kernel Methods

Spline Model Overview, Regression Splines, Smoothing Splines

STAT/BIOSTAT 527, University of Washington
Emily Fox
April 8th, 2014

Moving Beyond Linearity

- So far we have assumed standard linear models
- In the case of many predictors relative to number of observations, we considered penalized regression to avoid overfitting
- Often a convenient form, and necessary to assume simple structure to avoid overfitting in data-scarce regimes, but linear assumption rarely holds in practice

Moving Beyond Linearity

- Consider generic functional forms (univariate *x* for now)
  - If constrained to linear forms →
  - If arbitrary →
- As before, penalize complexity. Here, in terms of roughness.
  - If $\lambda \rightarrow 0$,
  - If $\lambda \rightarrow \infty$,
- Remarkable result: Explicit, finite-dimensional minimizer

Backtrack a bit…

- Instead of just considering input variables *x* (potentially mult.), augment/replace with transformations = "input features"

- ***Linear basis expansions*** maintain linear form in terms of these transformations

$$f(x) = \sum_{m=1}^{M} \beta_m h_m(x)$$

- What transformations should we use?
  - $h_m(x) = x_m$ →
  - $h_m(x) = x_j^2, \quad h_m(x) = x_j x_k$ →
  - $h_m(x) = I(L_m \leq x_k \leq U_m)$ →
  - …

Piecewise Polynomial Fits

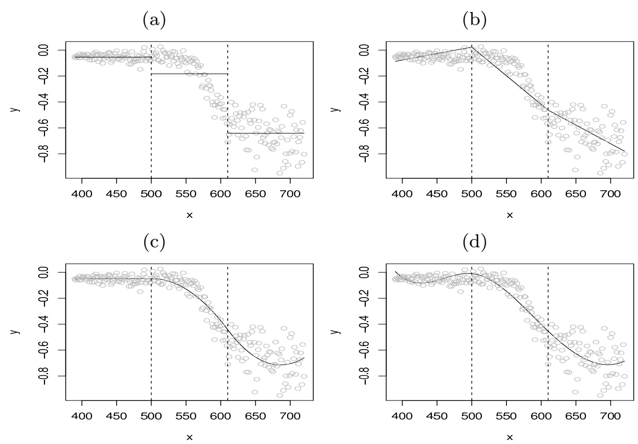
- Again, assume *x* univariate

- Polynomial fits are often good locally, but not globally
  - Adjusting coefficients to fit one region can make the function go wild in other regions

- Consider ***piecewise polynomial*** fits
  - Local behavior can often be well approximated by low-order polynomials

Piecewise Polynomial Fits

LIDAR Data Example



From Wakefield book

Piecewise Constant/Linear Fits

- **Example 1: Piecewise constant, with 3 basis functions**

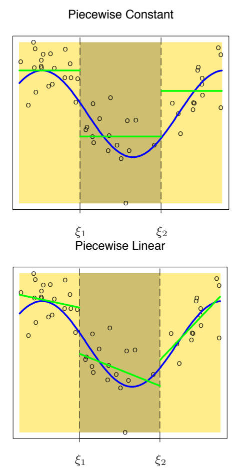
$h_1(x) =$

$h_2(x) =$

$h_3(x) =$

- Resulting model: $f(x) = \sum_{m=1}^{3} \beta_m h_m(x)$
- Fit: Take mean of data in each region
- **Example 2: Piecewise linear**
- Add three basis functions:

$$h_{m+3} = h_m(x)x$$



From Hastie, Tibshirani, Friedman book

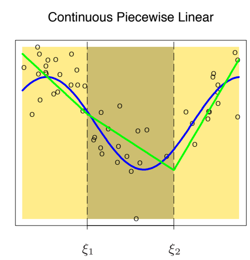
Regression Splines – Linear

- Resulting piecewise linear model:

$$f(x) = I(x < \xi_1)(\beta_1 + \beta_4 x) + I(\xi_1 \leq x < \xi_2)(\beta_2 + \beta_5 x) + I(\xi_2 \leq x)(\beta_3 + \beta_6 x)$$

  - # of params?

- Typically prefer continuity...
  - Enforce
  - Which implies
  - # params?

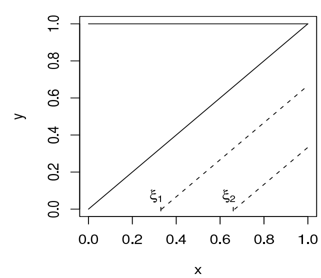


From Hastie, Tibshirani, Friedman book

Regression Splines – Linear

- More directly, we can use the ***truncated power*** basis

$$h_1(x) = 1$$
$$h_2(x) = x$$
$$h_3(x) = (x - \xi_1)_+$$
$$h_4(x) = (x - \xi_2)_+$$



From Wakefield book

- Resulting model:

- Continuous at the knots because all prior basis functions are contributing to the fit up to any single *x*

Regression Splines – Cubic

- Naively, extend as

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 (x-\xi_1)_+ + \beta_4 (x-\xi_1)_+^2 + \beta_5 (x-\xi_2)_+ + \beta_6 (x-\xi_2)_+^2$$

- But, 1st derivate is discontinuous (check this)
- Drop the truncated linear basis:

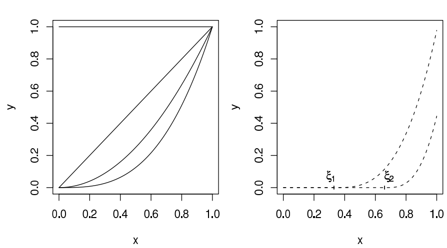
- Has continuous 1st derivative (check), but not 2nd

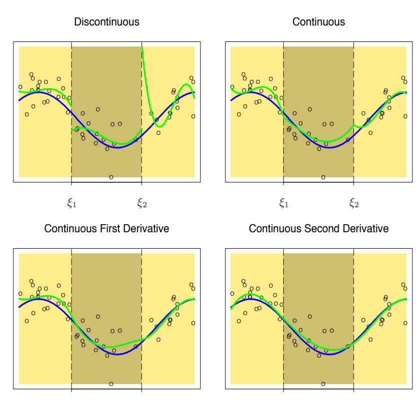
- Popular to consider ***cubic spline***:

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + b_1 (x-\xi_2)_+^3 + b_2 (x-\xi_2)_+^3$$

- Has continuous 1st and 2nd derivatives
- Typically people stop here

Cubic Spline Basis and Fit





Cubic Splines as Linear Smoothers

- Cubic spline function with *K* knots:

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \sum_{k=1}^{K} b_k (x - \xi_k)_+^3$$

- Simply a linear model

- Estimator:

- Linear smoother:

Natural Cubic Splines

- For polynomial regression, fit near boundaries is erratic.
  - Problem is worse for splines: each is fit locally so no global constraint
- ***Natural cubic splines*** enforce linearity beyond boundary knots
- Starting from a cubic spline basis, the natural cubic spline basis is

$$N_1(x) = 1 \qquad N_2(x) = x \qquad N_{k+2}(x) = d_k(x) - d_{K-1}(x)$$

$$d_k(x) = \frac{(x - \xi_k)_+^3 - (x - \xi_K)_+^3}{\xi_K - \xi_k}$$

- Derivation

Regression Splines – Summary

- Definition:

  *An **order-M spline** with knots* $\xi_1 < \xi_2 < \cdots < \xi_K$ *is a piecewise M-1 degree polynomial with M-2 continuous derivatives as the knots*

  *A spline that is linear beyond the boundary knots is called a **natural spline***

- Choices:
  - Order of the spline
  - Number of knots
  - Placement of knots

Return to Smoothing Splines

- Objective:

$$\min_f \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int f''(x)^2 dx$$

- Solution:
  - ***Natural cubic spline***
  - Place knots at every observation location x_i

- Proof: See Green and Silverman (1994, Chapter 2) or Wakefield textbook

- Notes:
  - Would seem to overfit, but penalty term shrinks spline coefficients toward linear fit
  - Will not typically interpolate data, and smoothness is determined by λ

Smoothing Splines

- Model is of the form: $f(x) = \sum_{j=1}^{n} N_j(x)\beta_j$
- Rewrite objective:

$$(y - N\beta)^T(y - N\beta) + \lambda\beta^T\Omega_N\beta$$

- Solution:
- Linear smoother:

Splines – Summary

- **Regression splines:**
  Fewer number of knots and no regularization
- **Smoothing splines:**
  Knots at every observation and regularization (smoothness penalty) to avoid interpolators

Reading

- Hastie, Tibshirani, Friedman: 5.1-5.5 (skipping 5.3)
- Wakefield: 11.1.1-11.2.3

What you should know…

- Linear basis expansions
- Regression splines
  - Cubic splines, natural cubic splines, …
  - Interpretation as a linear smoother
  - Degrees of freedom
- Smoothing splines
  - Arising from penalized regression setting with smoothness penalty
  - Cubic spline basis with knots at every data point