Module 1: Nonparametric Preliminaries

Task 1: Regression

- Assume a sample \((x_1, y_1), \ldots, (x_n, y_n)\)
- Model: \(y_i = f(x_i) + \epsilon_i\) \(E[\epsilon_i] = 0\)

- Task involves estimating the function \(f\)
- Goals of nonparametric approach:
  - Make few assumptions about \(f\)
  - Use a large number of parameters, but constrained in some way to avoid overfitting the data
  - Complexity can grow with the sample size
Parametric Regression

- **Parametric** inference assumes parametric form for $f(x)$
  
  $f(x) = \beta^T x$

  e.g., $f(x)$ is indexed by param. $\beta$

- Advantages:
  - Efficient estimation
  - Concise summarization

  e.g., LS est. of $\beta$, $\hat{\beta}_n$, leads to an est. of $f$

- What is the right parametric form for $f(x)$?
  Should it change w/ sample size?

Model Complexity

- How complex of a function should we choose?
  - To increase flexibility, using many parameters is attractive
    - Reduce bias
  - However, wide prediction intervals…
    - Fixed dataset contains a limited amt. of info
  - Leads to wild predictions
Example: Polynomial Regression

- For added flexibility, allow for high order polynomial, right?

\[ y_i = \sum_{j=0}^{p} b_j x_i^j + \epsilon_i \]

Not always good to add parameters

high bias, low var

Example: Polynomial Regression

- For added flexibility, allow for high order polynomial, right?

Sensitive to small changes in data

High order = low bias, but high var

How do we assess an estimator \( \hat{f} \)?

low bias, high var
Measuring Predictive Performance

- Having chosen a model, how do we assess its 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

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

- Assume $L_2$ loss
- Averaging over repeat training sets $Y_n = Y_1, \ldots, Y_n$ we get the **predictive risk** at $x^*$

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

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

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

- Finally, let’s average over covariates $x$

  - **Integrated MSE**
    \[
    \int \text{MSE}(\hat{f}_n(x)) p(x) \, dx
    \]
    summary over all inputs

  - **Average MSE**
    \[
    \frac{1}{n} \sum_{i=1}^{n} \text{MSE}(\hat{f}_n(x_i))
    \]

- Note: \( \text{avg. pred. risk} = \sigma^2 + \text{avg. MSE} \)

  - Monte Carlo est:
    \[ x_i \sim p \quad i = 1, \ldots, n \]

Bias-Variance Tradeoff

- Minimizing risk = balancing bias and variance

- Note: \( f(x) \) is unknown, so cannot actually compute MSE
In Practice…

- Minimizing risk = balancing bias and variance

![Graph showing model complexity vs. prediction error](image)

From Hastie, Tibshirani, Friedman

More on Nonparam Regression

- Often framed as learning functions with a complexity penalty
  - Regular behavior in small neighborhoods of the input
  - E.g., locally linear or low-order polynomial...estimator results from averaging over these local fits

- Choice of neighborhood = strength of constraint
  - Large neighborhood can lead to linear fit (very restrictive) whereas small neighborhoods can lead to interpolation (no restriction)
More on Nonparam Regression

- Different restrictions lead to different nonparametric approaches
  - Roughness penalty → splines
  - Weighting data locally → kernel methods
  - Etc.

- Each method has associated smoothing or complexity param
  - Magnitude of penalty
  - Width of kernel (defining “local”)
  - Number of basis functions
  - …

- Bias-variance tradeoff

- Will explore methods for choosing smoothing parameters

Reading

- Wakefield: 10.3-10.4
- Hastie, Tibshirani, Friedman: 7.1-7.3
What you should know

- What to report when data-generating mechanism is:
  - Known (optimal prediction)
  - Unknown and constrained to a specified model + loss fcn

- Example loss functions for
  - Continuous RVs
  - General RVs

- Goals of parametric vs. nonparametric methods

- Bias-variance tradeoff

- Measures of performance of estimators

Module 1: Nonparametric Preliminaries

Review of Regression, Linear Smoothers

STAT/BIOSTAT 527, University of Washington
Emily Fox
April 3rd, 2014
fMRI Prediction Subtask

- **Goal:** Predict semantic features from fMRI image

![Features of word](image)

**Linear Regression – review**

- **Model:**
  \[ y_i = \sum_{j=1}^{p} \beta_j x_{ij} + \epsilon_i \quad \text{for } i = 1, \ldots, n \]
  \[ = x_i^T \beta + \epsilon_i \]
  \[ E(\epsilon_i) = 0, \quad \text{var}(\epsilon_i) = \sigma^2 \]
  \[ x_{i1} = 1 \] for intercept

- **Design matrix:**
  \[ X = \begin{pmatrix}
  x_{11} & x_{12} & \cdots & x_{1p} \\
  \vdots & \vdots & \ddots & \vdots \\
  x_{n1} & x_{n2} & \cdots & x_{np}
  \end{pmatrix} \]

- **Rewrite in matrix form:**
  \[ (y_1, \ldots, y_n) \quad y = X \beta + \epsilon \]
### Least Squares

- **Least squares estimation:**
  - Minimize residual sum of squares
    \[ \hat{\beta} = \arg\min_{\beta} \left( y - X\beta \right)^T \left( y - X\beta \right) = \sum_{i=1}^{n} \left( y_i - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 \]
    \[ \frac{1}{2} \text{RSS}(\beta) = \frac{1}{2} \beta^T (X^T X) \beta - \beta^T X^T y + \text{const. wrt } \beta \]
  - Take gradient and set = 0
    \[ \nabla_{\beta} \frac{1}{2} \text{RSS}(\beta) = (X^T X) \beta - X^T y = 0 \]
    \[ \Rightarrow \hat{\beta}^{LS} = (X^T X)^{-1} X^T y \]

- In Gaussian case, LS est. = maximum likelihood est.

### Fitted Values

- **Fitted values**
  \[ \hat{\beta}^{LS} = (X^T X)^{-1} X^T y \]
  \[ \hat{f}_n = X \hat{\beta}^{LS} = Ly \]

- **Number of parameters**
  \[ p = \text{tr}(L) \]
  \[ \text{tr}(X(X^T X)^{-1} X^T) = \text{tr}(X^T X)^{-1} \text{tr}(X^T X) = \text{tr}(I_p) \]

- For any \( x \), we can write
  \[ \hat{f}_n(x) = l(x)^T y = \sum_{i=1}^{n} l_i(x) y_i \]
  where \( l(x) = x (X^T X)^{-1} X^T \) all \( x_{ij} \)'s from training data
Linear Smoothers

- Definition: \( \hat{f}_n \) of \( f \) is a **linear smoother** if, for each \( x \), there exists
  \[
  \ell(x) = (\ell_1(x), \ldots, \ell_n(x))^T
  \]
  such that
  \[
  \hat{f}_n(x) = \sum_{i=1}^n \ell_i(x)y_i
  \]

- Matrix form
  - Fitted values
  - Smoothing or "hat" matrix
  - Effective degrees of freedom:
    \[
    \gamma = tr(L)
    \]

Note 1:

A linear smoother does **not** imply that \( f(x) \) is linear in \( x \)

Note 2:

If \( Y_i = c \) for all \( i \), then \( \hat{f}_n(x) = c \) for all \( x \)

Solved during SLAM quiz

©Emily Fox 2014
fMRI Prediction Subtask

- **Goal:** Predict semantic features from fMRI image

\[ \hat{\beta} \times = (X^T X)^{-1} X^T y \]

- Rank Deficient

\[ n \gg p \]

- # training examples

\[ p = \# \text{voxels} = 70,000 \]
Regularization in Linear Regression

- Overfitting usually leads to very large parameter choices, e.g.:
  
  \[-2.2 + 3.1 X - 0.30 X^2\]  
  \[-1.1 + 4,700,910.7 X - 8,585,638.4 X^2 + \ldots\]

- Regularized or penalized regression aims to impose a "complexity" penalty by penalizing large weights
  
  "Shrinkage" method

Ridge Regression

- Ameliorating issues with overfitting:
  
  New objective:

  \[
  \min_{\beta} \sum_{i=1}^{n} (y_i - (\beta_0 + \beta^T x_i))^2 + \lambda \|\beta\|_2^2
  \]

  \[
  \min \text{RSS}(\beta) \quad \text{s.t.} \quad \|\beta\|_2^2 \leq S
  \]
Ridge Regression

- New objective:
  \[ \hat{\beta}_{ridge} = \arg \min_{\beta} \sum_{i=1}^{n} (y_i - (\beta_0 + \beta^T x_i))^2 + \lambda ||\beta||_2 \]

- Reformulate:
  \[ \frac{1}{2} F(\beta) = \frac{1}{2} \beta^T (X^T X) \beta - \beta^T X^T y + \text{const} \]

- Set gradient = 0
  \[ \hat{\beta}_{ridge} = (X^T X + \lambda I)^{-1} X^T y \]

- Linear smoother!!
  \[ f_{ridge} = X \hat{\beta}_{ridge} = L y \quad L = X(X^T X + \lambda I)^{-1} X^T \]

Ridge Regression

- Solution is indexed by the regularization parameter \( \lambda \)
- Larger \( \lambda \)  
  - high reg.
- Smaller \( \lambda \)  
  - low reg.
- As \( \lambda \to 0 \)  
  - \( \hat{\beta}_{ridge} \to \hat{\beta}_{ls} \)
- As \( \lambda \to \infty \)  
  - \( \hat{\beta}_{ridge} \to 0 \)

\[ \hat{\beta}_{ridge} = \arg \min_{\beta} \sum_{i=1}^{n} (y_i - (\beta_0 + \beta^T x_i))^2 + \lambda ||\beta||_2 \]
Shrinkage Properties

\[ \hat{\beta}_{\text{ridge}} = (X^T X + \lambda I)^{-1} X^T y \]

- If orthogonal covariates: \[ X^T X = I \]
  \[ \hat{\beta}_{\text{ridge}} = \frac{\hat{\beta}_{\text{ls}}}{1 + \lambda} v_j \]

- Effective degrees of freedom:
  \[ \nu = \text{tr}(L) = \text{tr}(X (X^T X + \lambda I)^{-1} X^T) \]

Ridge Coefficient Path

Typical approach: select \( \lambda \) using cross validation (CV)
A Bayesian Formulation

- Consider a model with likelihood
  \[ y_i \mid \beta \sim N(\beta_0 + x_i^T \beta, \sigma^2) \]
  and prior
  \[ \beta \sim N\left(0, \frac{\sigma^2}{\lambda} I_p\right) \]
- For large \( \lambda \)
  - Prior peaked around \( \beta = 0 \)
  - Penalising \( \beta \) far from 0
- The posterior is
  \[ \beta \mid y \sim N\left(\hat{\beta}^{\text{ridge}}, \sigma^2(X^TX + \lambda I)^{-1}X^TX\sigma^2(X^TX + \lambda I)^{-1}\right) \]
  \[ \hat{\beta}^{\text{MAP}} = \hat{\beta}^{\text{ridge}} \]

Variable Selection

- Ridge regression: Penalizes large weights
- What if we want to perform “feature selection”?
  - E.g., Which regions of the brain are important for word prediction?
  - Can’t simply choose predictors with largest coefficients in ridge solution
  - Computationally impossible to perform “all subsets” regression
  - Stepwise procedures are sensitive to data perturbations and often include features with negligible improvement in fit
- Try new penalty: Penalize non-zero weights
  - Penalty:
    \[ \ell_1 \mid \beta \mid_1 = \sum_j \mid \beta_j \mid \]
  - Leads to sparse solutions
  - Just like ridge regression, solution is indexed by a continuous param \( \lambda \)
LASSO Regression

- **LASSO**: least absolute shrinkage and selection operator

- New objective:

\[
\min_{\beta} \sum_{i=1}^{n} \left( y_i - (\beta_0 + \beta^T x_i) \right)^2 + \lambda \| \beta \|_1
\]

\[
\implies \min_{\beta} \text{RSS}(\beta) \quad \text{s.t.} \quad \| \beta \|_1 \leq B
\]

LASSO Solutions

- The LASSO solution is **nonlinear** in \(y\)... **not a linear smoother**
  - Degrees of freedom cannot be computed as before
  - Many recent studies on this (e.g., Zou et al. 2007, Tibshirani & Taylor 2011)
  - Standard errors via the bootstrap

- Efficient algorithms exist for solving
  - Will return to this next lecture
Reading

- Hastie, Tibshirani, Friedman: 3.2 (up to 3.2.3), 3.4
- Wasserman: 5.2

What you should know

- Linear regression
  - Least squares solution
  - Fitted values
- Definition of a linear smoother
- Ridge objective
  - L2 penalized regression solution
- LASSO objective
- Intuition for differences between ridge and LASSO solutions