Elements of Statistical Learning

book by Hastie, Tibshirani, and Friedmann

Chapter 1

LRS - Important stuff

1.1 Questions

When he writes f in ch2, is he referring specifically to the regression function $f(x) = E(Y|X = x)$ - or just any function? Is there a difference here, since the regression function, without any details on the joint distribution function is a just any function.

- What is the real content behind the "additive error assumption"? I.e., when he says, assume:

  $$Y = f(x) + \epsilon$$

  where $E(\epsilon) = 0$ and $Var(\epsilon) = \sigma^2$.

  Is f here the *regression function* $f(x) = E(Y|X = x)$ by assumption? (it seems to be a consequence at the very least)

  What is the Expectation in $E(\epsilon) = 0$ taken over?

  What else is hidden in this assumtpion? how general or restrictive is it?

  How is the joint probability distribution $Pr(X, Y)$ information encoded in here? And should ϵ be considered as another random variable.

- I like the idea of many variables $x_1, \cdots x_v$ with independent prob densities, and Y exactly determined by the vars. But then not measuring many variables, and hiding all their effects on Y in a single ϵ r.v. - so really we have (X, ϵ) as random variables.

  develop this further....

Chapter 2

Overview of Supervised Learning

2.0 LRS Summary of Main Ideas

- have observables (X, Y)
- want to predict Y based on observations of X
- given an observation of x, best predictor for Y (where best is defined via squared error loss function) is $f(x) = E(Y|X = x)$
- I assume this conditional expectation is defined using generally unkown joint probability distribution $Pr(X, Y)$
- ultimately we are trying to find a useful approximation for f
- The class of nearest neighbor methods can be viewed as a direct approximation to this conidtional expectation

  suffers from the curse of dimensionality
- talks about linear regression, a different class of model

  no curse of dimensionality - but potentially high bias
- generally most pairs (X, Y) will not have a deterministic relationship like $Y = f(X)$

  other unmeasured variables that also contribute to Y, including measurement error.
- often assume $Y = f(X) + \epsilon$ where $E(\epsilon) = 0$, there errors are independent of X, and are indentically distributed

  I am still uncertain as to its significance and import of this assumption

  I guess generally all you have for sure is a conditioanl distribution $P(Y|X = x)$

$$f_{Y|X}(y|x) = \frac{f_{X,Y}(x, y)}{f_X(x)}$$

  where f_X is the marginal density

$$f_X(x) = \int_y f(x, y)dy$$

- discusses two paradigms for obtaining said approximation $\hat{f}$ to f, given some data; the algorithmic gray box view, vs the geometric functional approximation view

  I believe it is fruitfull to label the approximation to $f(x)$ and $\hat{f}(x; D)$; It acknowledges that you need both a procedure to get the approximation, and some data.
- He then talks about the **complexity** of models and the bias variance tradeof.

  important point in all these proofs is to consider a single prediction point x_0 and a set τ of different (all possible?) training data sets D.

  in the additive error model this encompasses both the observed x's and the not directly observed ϵ, which he typically just uses $Y = f + \epsilon$ assumption to separate the sources of error.

  I this part should be done more clearly.

2.1 Notation

- Input variable typically denoted by X.
- if X is vector, its components accessed by subscript X_j
- upper case refers to generic/abstract variable. Observed values are written with lowercase: i.e. ith observed value is x_i (which again can be scalar or vector)
- Matrices represented by bold upper case $\mathbf{X}$

  example: a set of N input p-vectors x_i where $i = 1..N$, is represented as the $N \times p$ matrix X
- In general vectors are not bolded, except when they have N components. This distinguishes a p-vector of inputs x_i for the ith observation from the N-vector $\mathbf{x_j}$ consisting of all observations of variable X_j.
- All vectors are assumed to be column vectors. Hence the ith row of $\mathbf{X}$ is x_i^T.

2.2 Two Simple Approaches to Prediction: Least Squares and Nearest Neighbors

2.2.1 Linear Model

- Given a vector of inputs $X^T = (X_1, \cdots, X_p)$, we predict the output Y via

$$\hat{Y} = \beta_0 + \sum_{j=1}^{p} X_j \hat{\beta}_j$$

- often convenient to include the **bias** (aka intercept) term β_0 into X by including a constant variable 1 in X. Then letting $\hat{\beta}$ be the vector of coefficients including the bias, we can write the linear model in vector form as the inner product

$$\hat{Y} = X^T \hat{\beta}$$

- in general, $\hat{Y}$ could be a k-vector (i.e. the output is not scalar valued), in which case $\hat{\beta}$ would be a $p \times K$ matrix of coefficients.
- most popular approach to fit the linear model is the method of **least squares**: Pick the coefficients $\hat{\beta}$ to minimize the residual sum of squares:

$$RSS(\beta) := \sum_{i=1}^{n} (y_i - x_i^T \beta)^2$$

  in matrix notation

$$RSS(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta)$$

  for nonsingular $X^T X$ solution is

$$\hat{\beta} = \left(X^T X\right)^{-1} X^T Y$$

2.3 Nearest-Neighbor Methods

- The k-nearest neighbor fit for $\hat{Y}$ is defined as follows:

$$\hat{Y} = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$$

  where $N_k(x)$ is the neighborhood of x defined by the k closest points x_i in the training sample.

  requires metric to define closest - typically euclidean

  No training required! That is no parameters are fit.
- while k-NN appears to have a single parameter k, in truth it has N/k effective parameters, where N is data size.

  this is generally much larger than linear model parameters and thus requires much more data.

  heuristic: if you have nonoverlapping clusters ok k points, then you have N/k such nhoods, and thus need N/K parms (the means) to describe result/fit.
- cannot use RSS on training to pick k, because it would always pick $k = 1$, which leads to zero training error.

2.4 Statistical Decision Theory

- let $X \in \mathbb{R}^p$ be real valued random input vector, $Y \in \mathbb{R}$ be real valued random ouput, with joint distribution $Pr(X, Y)$
- we seek function $f(X) : \mathbb{R}^p \to \mathbb{R}$ for prediting Y value
- for this we need a **loss function** $L(Y, f(X))$ for penalizing errors in prediction
- most common loss function is: **squared error loss**

$$L(Y, f(X)) = (Y - f(X))^2$$

- criterion for choosing f: minimize the expected prediction error (EPE)

$$EPE(f) = E(Y - f(X))^2 = \int [y - f(x)]^2 Pr(dx, dy)$$

- He presents some short argument that via conditioning on X and pointwise minimization, you can arrive at the solution

$$f(x) = E(Y|X = x),$$

that is, the *conditional expecation*, aka the **regression function**.

- I am interested in how to rigorously solve this - perhaps a constrained variational calculus problem?
- KNN is an approximation to this, where instead of relying only on observations at x exactly, a neighborhood of x is use to obtain the expected value of y.
- for linear regression - we propose ansatz $f(x) \sim x^T\beta$ - so don't search over all functions

  this is a model based approach

  theoretical solutions is:

$$\beta = \left[E(XX^T)\right]^{-1} E(XY)$$

- Using L1 instead of L2 as loss function leads to $\hat{f}(x) = median(Y|X = x)$
- For categorical output, loss function is matrix $K \times K$ matrix L, where K is number of categories. Zero in the diagonals, and nonnegative elsewhere

  typical is **zero-one loss function** - where all off diagonals are 1.

- The $EPE = E[L(G, \hat{G}(x)]$ where expectation is taken over $Pr(G, x)$
- using same conditioning argumetn and pointwise minimization, and zero one loss, leads to solution known as the **Bayes classifier**

$$\hat{G} = g_k \quad \text{if} \quad Pr(g_k|X = x) = max_{g \in G} Pr(g|X = x)$$

that is, classify as the most probable class, using discrete conditional distribution $Pr(G|X)$.

Error rate of the bayes classifier is often called the **Bayes rate**.

2.5 Local Methods in High Dimension

- Fitting/prediciton methods which rely on local approximations (like KNN), struggle as dimensions get high - **the curse of dimentionality**
- example: consider p-dimensional unit hypercube with uniformly distributed inputs. If we place a sub hypercube about the origin, such that a fraction r of the data is contained there, the linear dimension of this hypercube must be $r^{1/p}$. For 10 dimensions, the expected edge length for $r = 0.01$ is 0.63.

  so to capture one percent of the data, you must go %63 of the distance avaialble in each dimension! That is not local.

- another manifestation comes from looking at unit sphere with uniform distribution - median value for closest to origin is about 1/2 radius. So every point close to edge.

  this is a problem because for points on the edge, prediction is often extrapolation instead of interpolation. Which is much shakier.

- another manifestation of this curse is that the sampling density is proportional to $N^{1/}$ where N is sample size and p is dimension. So if in 1-D $N = 100$ represents a dense sample size, the equivalent density in 10 D is 100^{10}.

  so in high D, all feasable samples are sparse.

- **Bias-variance tradeoff** - nice little "proof" - setup: fix a point in domain x_0, get a large *set of training samples* τ - study the expected error in prediction at x_0 that comes from the sampling.

  notation: $\hat{y}_0$ is prediction using some model. y_0 is true value.

$$\begin{aligned} MSE(x_0) &= E_\tau[(y_0 - \hat{y}_0)^2] && (2.1)\\ &= E_\tau[(y_0 - E_\tau[\hat{y}_0] + E_\tau[\hat{y}_0] - \hat{y}_0)^2] && (2.2)\\ &= E_\tau[(y_0 - E_\tau[\hat{y}_0])^2] + 2E_\tau[(y_0 - E_\tau[\hat{y}_0])(E_\tau[\hat{y}_0] - \hat{y}_0)] + E_\tau[(\hat{y}_0 - E_\tau[\hat{y}_0])^2] && (2.3)\end{aligned}$$

  The term in blue is the variance of the prediction. The term in orange: $E_\tau[(y_0 - E_\tau[\hat{y}_0])^2] = (y_0 - E_\tau[\hat{y}_0])^2$ which is just the Bias of prediction squared. The middle term (in black) is zero (can factor out first piece, and distribute expetation over subtraction). So

$$MSE(x_0) = Bias_\tau(\hat{y}_0)^2 + Var_\tau(\hat{y}_0) \quad (2.4)$$

  This relationship ends up being very generic. More on this at the end of the chapter.

2.6 Statistical Models, Supervised Learning and Functional Approximation

- goal is to fint approximation $\hat{f}$ to the function f that undeliers the predictive relationship between inputs and outputs
- Sum of squared errors loss leads to optimal f being $f(x) = E(Y|X = x)$ - called the **regression function**.
- The class of nearest neighbor methods can be viewed as a direct approximation to this conidtional expectation
- More generally, we seek to approximate conditional probability $Pr(Y|X)$
- one common model: **additive error model**

  assumes:

$$Y = f(X) + \epsilon$$

  where f is the regression function, ϵ is a random error independent of X with some distribution such $E(\epsilon) = 0$

  in what sense is error dependent or independent of Y? - clearly for a given x, error has perfect correlation to Y - excess y is the error. But what about in general? Is that even a sensible question to ask?

  McElreath would say this is terrible way to think about it - better to just say something like $Y \sim N(\mu_x, \sigma)$ and $\mu_x = f(x)$ - this generalized better to non additive models. The additive nature in this case evident from normal distro.

- MAIN POINT: This is a specific claim about the conditional probability distribution $Pr(Y|X)$, namely that Y is distributed like ϵ plus a value determined by X.

  moreover, note that X only comes in through the conditional mean $f(x)$, and it does not come into the variance of Y (though that can and is often relaxed)

- For quantitative responses, this is typically not the assumption, but rather that of some bernoulli process (for binary var) with p of outcomes determined by X, that is $p(X)$. This binds both the conditional expectation and the variance to x.

  McElreath would say $Y \sim Bern(p_x)$ and $p_x = P(X)$.

- His claim is that there are two main ways to think of this endeavor to find a good approximation for f
  - Supervised learning - there is some algorithm that can take an input x_i and map it to an output $\hat{f}(x_i)$, which can also adjust $\hat{f}$ based on the difference between predicted value and observed y_i. This algorithm should produce a map that can be used for predictions.
  - Function Approximation - x_i, y_i are viewed as points in a (p+1)-dimensional Euclidean space. The idea is that the data satisfies some relationship $y_i = f(x_i) + \epsilon_i$, and goal is to obtain a useful approximation to f thatn is valid for all points in some region.

    This paradigm encourages mathematical concepts of geometry and probability, so they prefer it.

- often the approximations are restricted to some parameterized fammily of functions, and the challenge is to find the best parameter

  ex: linear model - $f(x) = x^T\beta$ (params are β), or more generally a **linear basis expansion**

$$f_\theta(x) = \sum_{k=1}^{K} h_k(x)\theta_k$$

  where h_k are a suitable set of functions or transformatiosn of the input vector.

- often fit by minimizing the residual sum-of-squares (this is just least squares error).
- A more general criteria is **maximum likelihood estimation** aka MLE
  - given random sample $y_i, i = 1 \cdots N$ from a density $Pr_\theta(y)$,
  - log-probability of data is
$$L(\theta) = \sum_i log(Pr_\theta(y_i))$$
  - the pricniple of MLE says most reasonable θ is that which maximizes $L(\theta)$
  - Least squares with additive error model $Y = f_\theta(x) + \epsilon$ with $\epsilon \sim N(0, \sigma^2)$ is equivalent to MLE using conditional likelihood $Pr(Y|X) = N(f_\theta(X), \sigma^2)$
  - Another example: Assume multinomial qualitative output G, with regression function $Pr(G|X)$. Suppose we have model $Pr(G = g_k|X = x) = p_{k,\theta}(x)$, then the loglikehood is
$$L(\theta) = \sum_{i=1}^{N} \log p_{g_i,\theta}(x_i)$$
    which is also referred to as the **cross-entropy**
- From entropy, $\int_x p_x log(p_x)$ to cross entropy $\int_x u_x log p_x$, then to observed cross entropy (the $\int_x u_x$ become the sum ovber observed values) $\sum_i logp(x_i)$

2.7 Structured Regression Models

- infinite many solutions to min RSS (they just have to interpolate between data somehow)
- must impose restrictions on family of potential solutions - they are controlling the *complexity* of solutions in one way or another

  often impose some regularity on small neighborhoods

  size of neighborhood dicatates strength of complexity reduction (in k-NN, k controls that)
- main point: any method that constraints local variation in small isotropic neighborhoods will suffer curse of dimensionality; any method that overcomes the curse has some way of measuring neighborhoods which does not allow them to be small in all directions.

2.8 Classes of Restricted Estimators

- main approaches listed below:
- Roughness penalty
$$PRSS(f; \lambda) = RSS(f) + \lambda J(f)$$
where J is some functional that will large for rapidly changing functions over small reguins.

  one example $J(f) = \int [f''(x)]^2 ds$. $\lambda = \infty$ only allows linear functions.

  these are also called **regularization** methods
- Kernel methods

  estimate the regression function by specyfing nature of local neighborhoods

  use a **kernel function** $K_\lambda(x_0, x)$ which assigns weights to points x in a region around x_0

  example, Gaussian density function
$$K_\lambda(x_0, x) = \frac{1}{\lambda} \exp\left(-\frac{||x - x_0||^2}{2\lambda}\right)$$

  one example is the Nadaraya-Watson weighted average, where $\hat{f}(x)$ is the weighted sum of all y_i observations times kernels $K_\lambda(x, x_i)$.

  In general we can define a *local regression estimate* of $f(x_0)$ as $f_{\hat{\theta}}(x_0)$, where $\hat{\theta}$ minimizes
$$RSS(f_\theta, x_0) = \sum_{i=1}^{N} K_\lambda(x_0, x_1)(y_i - f_\theta(x_i))^2$$
and f_θ is some parameterized function, like a low order poly

ex: $f_\theta(x) = \theta_0$, results in Ndaraya-Watson

or linear $f_\theta(x) = \theta_0 + \theta_1 x$ — results in popular **local linear regression model**

notice that RSS here depends on both f_θ and x_0

- Basis functions

  includes linear and polynomial expansions (and much more)

  postulate a structure

$$f_\theta(x) = \sum_{m=1}^{M} \theta_m h_m(x)$$

  note that it is linear in the θs.

2.9 Model Selectio and the Bias Variance tradeoff

- all of the models above has a **smoothing** or **complexity parameter**
- cannot use RSS on training data to determine these parameters

  such method ends up picking a solution that interpolates between data and hence has zero residuals, but is wild in between and not good at predicting future data.

- he goes through another - weirdly artifical - bias variance decomposition that show the dependency of the piece on the complexity parameter (for knn)
- main idea: as model complexity increases, squared bias decreases but variance increases.
- ideally, chose complexity parameter that leads to minimum test error.
- obvious estimate of test error is train error.
- unfortunately test error does not properly account for error that comes from model complexity.