Statistics 8330: Data Analysis III
Introduction

Suggested Reading: James, Witten, Hastie, and Tibshirani (JWHT), 2017, Chapters 1-2

Supplemental Reading: Hastie, Tibshirani, and Friedman (HTF), 2009, Chapters 1-2

Christopher K. Wikle

University of Missouri
Department of Statistics

Problems

Consider the following problems:

- Identify the risk factors for a particular type of disease, based on clinical and demographic variables.
- Identify the cues in vibration signals from a male insect that suggest mating success.
- Predict the profile of soil water content given depth profiles of electroconductivity at various wavelengths.
- Identify which proteins are most associated with a particular gene response.
- Predict the price of a stock in 6 months, based on recent performance and other economic data.
- Predict whether an email is spam or not.
- Identify the key subspace that defines climate variability for a particular region.
- etc.

What type of statistical model would you consider to answer such questions?

Another Realistic Problem: Influence of Coupons on Order Patterns

Assume we are working for an online shop with accompanying coupon generation. We are interested in the impact of coupons on the shopping basket value as well as the redemption rate of the individual coupons. Coupons have been used as purchase incentives for quite some time in a variety of businesses.

There are several questions, including:

- Who responds to coupons?
- Who would have made the purchase even without the coupon?

We must give equal consideration to both of these questions.

Thus, using historical order data from the online shop, we must create a model that comes up with a prediction for the redeemed coupons and for the shopping basket value for new orders with the shop.

How would you go about modeling this?

Stat 8310: DA I

CKW

What were you supposed to learn in Stat 8310?

- A **survey course** covering basic linear models (regression, ANOVA); basic inference; basic design (multi-factor; randomized block); basic fixed, random, mixed models; basic categorical data analysis; basic nonparametric methods
- Overview of applied statistical methods from 1880s - 1960s
  - univariate, linear, Gaussian (except for the categorical methods), independent errors

Could these methods be used to address the problems described above?

What were you supposed to learn in Stat 8320?

- A **survey course** covering applied statistical methods from about the 1930s - 1990s.
- Concerned with problems where the assumptions from Stat 8310 are violated
  - nonlinear response functions
  - non-Gaussian error structure
  - dependent errors
  - multivariate responses
- Nonlinear Regression, Generalized Linear Models, Mixed Models, Generalized Linear Mixed Models, Principal Component Analysis, Factor Analysis, Discriminant Analysis, Cluster Analysis

Again, could these methods be used to address the problems described above?

Data Analysis III

What are we going to do in DA III?

- A **survey course** covering some of the fundamental data methods in **statistical learning**.
- This is not a theory course and the focus will be on developing an understanding of the motivation and application of advanced data analysis methodologies that are being used to perform statistical "data mining" and "statistical machine learning."
- The course emphasizes computer application and data analysis of complex data sets such as one would find in the real world.
  - **"Big Data"**, "Complex Data"
- For the most part, this course will *not* cover non-statistical methods of data mining (Exception: "deep learning")

What is Statistical Learning?

- **supervised statistical learning:** "building a statistical model for predicting, or estimating, an output based on one or more inputs"
- **unsupervised statistical learning:** "building an understanding of relationships between inputs when no supervising output is available"

But, haven't we already done this in DA I and DA II?

Yes! (e.g., regression is supervised statistical learning and principal component analysis is unsupervised statistical learning).

So, what is different?

- More complicated (unknown) relationships!
- High volume of data
- Evaluation of models

Yet, as you will see, many of the fundamental concepts are the same as what we have learned in the previous courses.

General Modeling Framework

We will typically follow the notational conventions in JWHT(2017) and HTF(2009).

Let Y correspond to our output variable (response, dependent variable); typically, we will use Y for continuous output variables and G (sometimes) for categorical output variables.

Let X correspond to input variables (predictors, independent variables, "features"); usually, we have many (say, p) input variables, so $X = (X_1, \ldots, X_p)$.

In the supervised learning case, we are typically interested in a basic *additive error model*:

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

where $f(\cdot)$ is typically fixed but unknown and contains systematic information, and ϵ is a random error term (critically, assumed to be mean 0 and *independent of* X).

General Framework: Prediction

In cases where the focus is prediction, we don't really care which inputs are important or what is the form of $f(\cdot)$. Rather, we just want to be able to predict the output Y given new inputs. This requires that we estimate $f(\cdot)$ (sometimes, we call this a "black box"):

$$\hat{Y} = \hat{f}(X)$$

- **Reducible Error:** $\hat{f}(\cdot)$ is not a perfect estimate of f but it can be improved in principle (not necessarily in practice)
- **Irreducible Error:** ϵ is not a function of the inputs (by definition) so it can't be reduced (it may contain other unmeasured (or unknown) variables or individual variation that can't be controlled)

$$\begin{aligned} E[Y - \hat{Y}]^2 &= E[f(X) + \epsilon - \hat{f}(X)]^2 \\ &= [f(x) - \hat{f}(X)]^2 + \text{var}(\epsilon) \\ &= \text{reducible} + \text{irreducible} \end{aligned}$$

General Framework: Inference

In cases where the focus is on inference, we want to understand the *relationship* between X and Y or how Y changes as a function of $X_1, \ldots, X_p$ (or some subset of these input variables)

In this case we can't treat $\hat{f}(X)$ as a "black box." We may care about

- which predictors are associated with Y?
- what is the relationship between the response and the predictors (e.g., linear, nonlinear)?

In statistics, we often consider $f(X)$ to be the conditional expectation associated with the conditional distribution of $Y|X$ (recall linear regression), but it typically isn't known and must be estimated.

Estimating f

Assume we have n samples of training data available to estimate f.

- x_{ij}: value of the jth predictor or input for observation i, $i = 1, \dots, n; j = 1, \dots, p$; Let $x_i = (x_{i1}, x_{i2}, \dots, x_{ip})^T$ (note: T is the "transpose" operator)
- y_i: observed response for ith observation
- Training data set: $\mathcal{T} = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$

We seek $\hat{f}$ such that $Y \approx \hat{f}(X)$ for any observations (X, Y).

In general, we can take a *parametric* or *nonparametric* approach to estimating f.

Estimating f: Parametric Approach

Generally, there are two steps to a parametric approach:

1. Make an *assumption* about the functional form of $f(X)$: e.g.,

$$f(X) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$$

2. Fit (or "train") the model; consists of estimating the parameters (e.g., $\{\beta_0, \beta_1, \ldots, \beta_p\}$); e.g., use OLS in regression or MLE more generally

Parametric approaches are typically simpler because *we just have to estimate the parameters, not the functional form of f*. In addition, it is easy to interpret the parameters in this setting because they correspond to a known functional form.

However, these functions may be overly simplistic and the input/output relationship may require more flexible functional forms.

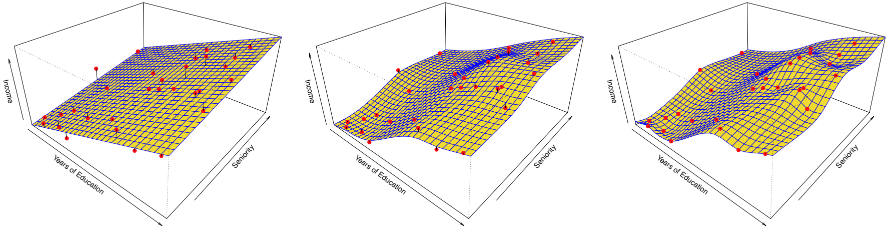
Estimating f: Nonparametric Approach

In the nonparametric case, we seek to estimate a more flexible f that is in some sense “close” to the data points but does not over fit. In general, nonparametric methods

- can fit more flexible structure
- typically require *many* more observations because we have to estimate f, not just the parameters
- can easily lead to over fitting (i.e., modeling the noise)
- are more difficult to interpret than parametric models

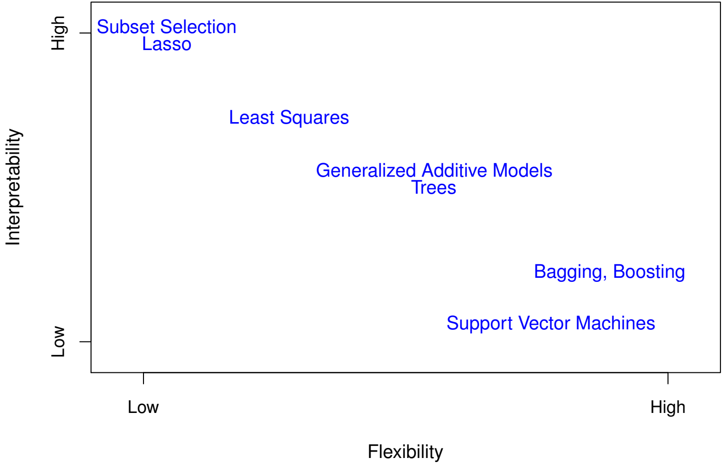
In general, there is always a tradeoff between prediction accuracy and model interpretability. Typically, restrictive models are more interpretable (and, thus used more for inference) and flexible models are less interpretable (and more often used for prediction).

Example Model Fits (from JWHT, Chap. 2)



Left: Linear function, Middle: Smooth thin plate spline, Right: Rough thin plate spline

Tradeoff Between Model Flexibility and Interpretability (JWHT, Chap. 2)



Regression vs Classification

Most supervised problems of interest in statistical (and machine) learning applications are broadly classified as either *regression* or *classification*. This should be no surprise to us, because we have done both in DA I and II. For example, Regression and ANOVA are essentially the same thing depending on whether data are from designed experiment or not; categorical regression (logistic) and discriminant analysis are classification methods. More generally, GLMs and GLMMs can be either depending on whether the response is categorical, and LMMs are regression problems with random effects.

Sometimes, the distinction between classification and regression is not clear.

In both cases, we need ways to assess model accuracy. This allows us to build better models and to choose between models.

Assessing Model Accuracy

Clearly, no one statistical method will always work best. Thus, we need ways to choose between methods and models. First, we need to develop measures of the quality of model fit.

Recall, in regression, we used the *mean squared error (MSE)*:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2,$$

where this is really the "*training MSE*" since it is based on the observations that were used to train the model. It would be better to base our evaluation on data not used to train the model (i.e., a "test" or "validation" sample).

Assessing Model Accuracy

That is, recall our training set, $\mathcal{T}$, was used to estimate $\hat{f}$, but we aren't really interested in $\hat{f}(x_i) \approx y_i, i = 1, \ldots, n$.

Rather, we are interested in $\hat{f}(x_0) \approx y_0$ for (x_0, y_0), a test observation not used to train the statistical learning method.

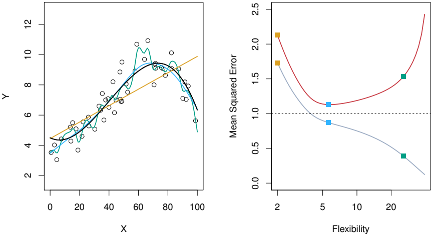
Thus, we seek the method with the lowest "*test MSE*" as opposed to training MSE; e.g., Ave$(\hat{f}(x_0) - y_0)^2$ for a large number of training observations.

Note, **generalization error** is a term in machine learning that refers to a measure of how accurately an algorithm or model is able to predict outcome values for data not used to train the model.

Assessing Model Accuracy (cont.)

Clearly, it is best to have a test data set, but this may not be available. In that case, why can't we just use training MSE?

There is no guarantee that the model with the smallest training MSE will give the smallest test MSE. This is related to the flexibility of the model. In particular, as the flexibility of the method increases, we observe a monotone decrease in the training MSE and a U-shape in the test MSE. Consider the following example.



Left: Data simulated from f. Three estimates of f: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves). Right: Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.

Assessing Model Accuracy (cont.)

Why does this happen? It is just a manifestation of the usual *bias-variance tradeoff* that we have seen a number of times in DA I and DA II. Recall, we can write the expected test MSE as:

$$E(y_0 - \hat{f}(x_0))^2 = \text{var}(\hat{f}(x_0)) + [\text{bias}(\hat{f}(x_0))]^2 + \text{var}(\epsilon)$$

Note that $E(MSE) \geq \text{var}(\epsilon)$ because ϵ is irreducible model error and its variance is a lower bound on the quality of our potential fit.

Flexible modeling approaches have more variance but less bias. Parametric approaches typically have less variance but more bias.

The U-shape in the MSE vs flexibility curve is present because as you move from less to more flexibility, *the bias decreases faster initially than the variance increases*, suggesting that the test MSE decreases. Eventually, the bias levels out but the variance continues to increase, leading to an increase in the test MSE.

Our big challenge: *find methods that balance the bias-variance tradeoff!*

Model Accuracy in the Classification Setting

MSE doesn't make much sense when we have categorical responses as in classification. We can develop other measures that are intuitive and have nice properties. In this case, y_i is considered to be a categorical response. (Note: HTF(2009) use g_i to represent categorical responses, but JWHT(2017) typically use y_i and make the distinction in context).

We could use the *training error rate*, which is just the proportion of mistakes in classification from applying $\hat{f}$ to the training set:

$$\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i),$$

where $I(y_i \neq \hat{y}_i)$ is an indicator variable that equals 1 if $y_i \neq \hat{y}_i$ and 0 otherwise.

Similarly, the *test error rate* is given by:

$$\text{Ave}(I(y_0 \neq \hat{y}_0)),$$

for test observation (x_0, y_0).

Classification Setting (cont.)

The test error rate can be shown to be minimized on average by a very simple classifier that assigns each observation to the most likely class, given its predictor values. This is just the conditional probability that $Y = j$ given the observed predictor x_0. This is known as a *Bayes classifier*.

Bayes classifier: assigns x_0 to class j where $Pr(Y = j|X = x_0)$ is largest.

Note, if there are only two classes, this corresponds to predicting into class 1 if $Pr(Y = 1|X = x_0) > .5$, and class 2 otherwise. We say that the *Bayes decision boundary* is given by $Pr(Y = 1|X = x_0) = .5$.

The Bayes classifier produces the lowest possible test error rate, which is called the *Bayes error rate.* In general, the overall Bayes error rate is given by

$$1 - E_x(\max_j[Pr(Y = j|X)]).$$

This is greater than 0 if the classes overlap, which is analogous to the irreducible error in the additive model.

Classification Setting (cont.)

The problem with applying the Bayes error rate in practice is that it requires knowing $Pr(Y|X)$, which we almost never do. Most classification methods end up providing some sort of estimate of this conditional probability.

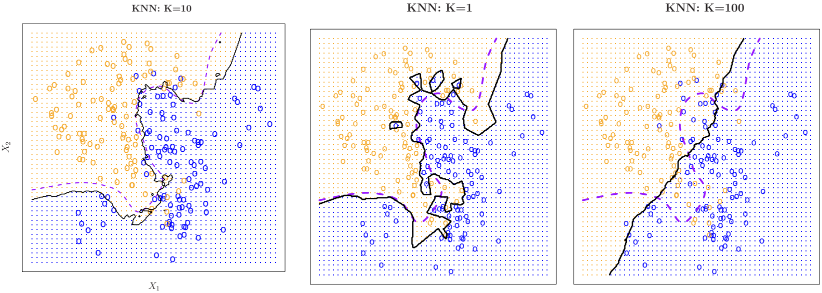
As an example, consider the *K-nearest neighbor (KNN)* classification approach that we saw in DA II. In this case,

$$Pr(Y = j|X = x_0) \approx \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(y_i = j),$$

where $\mathcal{N}_0$ represents the K points in the training data that are closest to x_0.

This can be close to the optimal Bayes classifier, but we still have to balance the bias-variance tradeoff (flexibility tradeoff) through the choice of the number of neighbors, K.

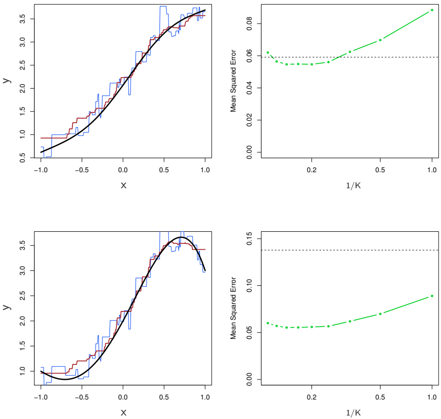
Classification Example (JWHT(2017), Chap. 2)



The black curve indicates the KNN decision boundary on the data from Figure 2.13, using K = 10. The Bayes decision boundary is shown as a purple dashed line. The KNN and Bayes decision boundaries are very similar.

A comparison of the KNN decision boundaries (solid black curves) obtained using K = 1 and K = 100 on the data from Figure 2.13. With K = 1, the decision boundary is overly flexible, while with K = 100 it is not sufficiently flexible. The Bayes decision boundary is shown as a purple dashed line.

Nearest Neighbor Regression (JWHT(2017), Chap. 3)



Top Left: Slightly non-linear relationship between X and Y (solid black line), the KNN fits with K = 1 (blue) and K = 9 (red) Top Right: For the slightly non-linear data, the test set MSE for least squares regression (horizontal black) and KNN with various values of 1/K (green). Bottom Left and Bottom Right: As in the top panel, but with a strongly non-linear relationship between X and Y.

Assessing Model Accuracy (cont.)

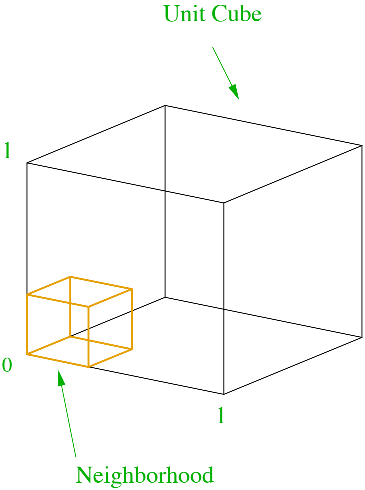
As we have seen, for both the regression and classification settings, assessing model accuracy can be a delicate balancing act between variance and bias – particularly, when we have to use only a training sample. Without a comprehensive test data set, we will typically use *cross-validation* methods to help do this (as discussed later).

Implicit in the preceding material is that in both the regression and classification setting we are seeking estimates of the conditional probability $Pr(Y|X)$ (recall that in classical linear regression, we obtain the mean and variance of this conditional distribution). We also are starting to notice that more "local" estimates (considering smaller neighborhoods) of this conditional distribution give more flexible models.

There is another critical issue that comes up when we are interested in neighbor-based methodologies. This has to do with the **curse of dimensionality**.

Curse of Dimensionality

Consider a p-dimensional hypercube of unit volume. e.g., consider a unit cube and a sub-cube neighborhood at the origin (HTF(2009), Chap. 2):



Curse of Dimensionality (cont.)

What would the length of a side of our sub-cube have to be to capture a fraction (say, r) of the observations? One can show that, on average, this is given by $e_p(r) = r^{1/p}$, where p is the dimension of the hypercube. E.g.,

$$e_1(.01) = 0.01, \;\; e_1(.1) = 0.1, \;\; , e_{10}(.01) = 0.63, \;\; e_{10}(.1) = 0.80$$

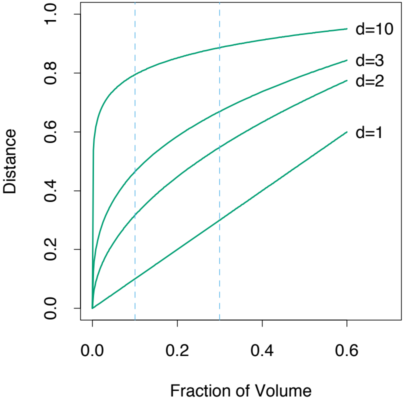
This is pretty remarkable: in 10 dimensions your sub-cube has to be over 60% of the entire sample space to cover (on average) just 1% of the observations!

This implies something *very important*: **there is no such thing as a “local” neighborhood in high-dimensions!**

In fact, the median distance in the unit hypercube from the origin to the closest point is given by $d(p,n) = (1 - \frac{1}{2}^{1/n})^{1/p}$. E.g.,
$d(p = 10, n = 500) \approx 0.52$, which implies that most data points are closer to the boundary of the cube than to any other point. This presents a problem for neighborhood-based methods!

Curse of Dimensionality Illustration (HTF(2009), Chap. 2)

The figure below shows the side-length of the sub-cube needed to capture a fraction r of the volume of the data for different dimensions, d.



Statistical Learning: Summary

- Machine learning tends to perform learning algorithmically; e.g., in the supervised case, finding an algorithm to select $\hat{f}$ such that $y_i - \hat{f}(x_i)$ is in some sense minimized relative to new observations.
- Statistical learning tends to focus more on function approximation and the associated uncertainties. There are many different ways to do this. One must consider:
  - trade-offs between parametric vs nonparametric approaches (e.g., functions vs neighbors)
  - trade-offs between interpretability vs prediction
  - trade-offs between variance and bias
  - the curse of dimensionality
  - different measures of model accuracy
  - different optimization criteria
- Typically, the statistical methods *penalize* some measure of the average difference $(y_i - f(x_i))$ to enforce some degree of smoothness or prior information; these are restricted estimators or broadly, regularization methods.

This Class

A big part of this class will be to explore these tradeoffs in various types of models and methods. We will not go into the details of these approaches in most cases, but will provide the proper motivation and implementation strategies. You are highly encouraged to read additional information about these methods and their implementation, as well as practice with various computing packages (in particular, using R).