36-724 Spring 2006: EDA, Prediction, Classification, Modeling

Brian Junker

March 20, 2006

- Some Statistical / Data-Analytic Activities
- Prediction/Classification
- Linear Regression and Nearest Neighbors
- Features and Questions
Some Statistical / Data-Analytic Activities

• Exploratory Data Analysis; Ad-hoc Dimension Reduction; “Unsupervised Learning”
  – Elementary numerical and graphical summaries
  – PCA/FA/ICA/etc.
  – Clustering
  – Projection Pursuit
  – MDS
  – etc. etc.

• Prediction and Classification; “Supervised Learning”
  – Linear and nonlinear regression models; additive models;
    projection-pursuit regression and neural networks; etc.
  – $k$-nearest-neighbors;
  – Discriminant analysis; tree-based methods (CART); support-vector classifiers; etc.

• Scientific Modeling; “Statistical Science”
  – Experimental (medicine, agriculture, engineering) and survey (social sciences) data
Scientific Modeling; “Statistical Science” [continued]

- Models often directly reflect the structure and assumptions of the data collection (“experimental design”) or an underlying data generation theory (Rasch model; choice models; survival models; etc.)

None of these are model-free! However they are ordered from “weakest model” to “strongest model”:

- EDA
- Prediction and classification
- Scientific modeling

The boundaries are not strict.

Different sorts of model complexity depending on purpose (probabilistic models such as Hierarchical Bayes, vs algorithmic models such as $K$-nearest-neighbors). Both can be computationally complex!

Different measures of success depending on purpose (summarization, inference, prediction, ...).
Prediction/Classification

Types of problems

• Predict whether a patient, hospitalized due to heart attack, will have another. Use demographic, diet, clinical covariates.

• Identify the risk factors for prostate cancer, based on clinical and demographic covariates.

• Predict the price of a stock six months from now based on recent performance of the company and the economy.

• Identify the numbers in a handwritten zip code, from digitized images.

• Score the quality of an essay.

• Classify incoming email as spam or not spam.
Types of data

• Data is typically *multivariate* with many *features* or *coordinates*.

• There may or may not be many *cases*.

• For supervised learning problems, the data is typically broken up into
  
  – *Training set*: We use both the covariates (predictor variables) and true outcomes or classifications, to train the model to make good predictions (model fitting).

  – *Testing set*: Use the covariates to make model-based predictions and compare to the true outcomes/classifications.

Dividing up the data into training/testing sets is an example of the *cross-validation paradigm*. It tries to explicitly control the “bias/variance tradeoff” implicit in most model fitting and prediction.

Different from intrinsic model-fit indices like $\chi^2$, likelihood ratio, Bayes factors. Penalized indices like AIC/BIC/etc. implicitly try to control “bias/variance tradeoff” (and are faster than cross-validation).
Linear Regression and Nearest Neighbors

Let $Y$ be a random variable representing outcomes and $X = (X_1, \ldots, X_p)^T$ be a random vector of predictors (covariates). We wish to construct a function $f(x)$ such that $Y \approx f(X)$ over the joint distribution $P(dx, dy)$.

It is useful to think about is the expected prediction error,

$$EPE(f) = E[(Y - f(X))^2] = \int (y - f(x))^2 P(dx, dy)$$  \hspace{1cm} (*)

(Squared error loss $(Y - f(X))^2$ is not the only kind we can think about but it is convenient here). By conditioning we see that

$$EPE(f) = E_X \{ E_{Y|X}[(Y - f(X))^2|X] \}$$

so, to minimize $EPE(f)$, it suffices to consider

$$f(x) = \arg\min_c E_{Y|X}[(Y - c)^2|X = x] = E[Y|X = x]$$

So prediction and classification amount to obtaining good estimates of the regression function $E[Y|X = x]$. 
Linear Regression

Linear regression tries to model \( f(x) = E[Y|X = x] \) by assuming that \( f(x) \approx x^T \beta \). Plugging this into (\( \ast \)) and differentiating to minimize over \( \beta \) we get that

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

As we know, solving the same problem for a data vector \( y = (y_1, \ldots, y_n)^T \) and data matrix \( X = [x_{ij}] \) with rows \( x_i^T \), and replacing the theoretical quantity (\( \ast \)) with

\[
S(\beta) = \sum_{i=1}^{n} (y_i - x_i^T \beta)^2
\]

(note the connection with statistical functionals in bootstrapping), we obtain the usual least-squares estimator

\[
\hat{\beta} = (X^TX)^{-1}X^Ty
\]

so that

\[
\hat{f}(x) = x^T \hat{\beta}
\]
**Nearest Neighbors**

The nearest-neighbor approach attempts to estimate \( f(x) = \text{E}[Y|X = x] \), using the training data directly.

With probability 1, there will be 0 or 1 observations at any given \( x \). Therefore we settle for “nearby” points where there are observations to work with.

At each point \( x \), we find the set of \( k \) nearest neighbors \( N_k(x) \). Then \( k \)-nn estimates

\[
\hat{f}(x) = \text{Ave}\{y_i \mid x_i \in N_k(x)\}
\]

This involves two approximations or relaxations:

- Expectation is relaxed to averaging over sample data
- Conditioning on \( X = x \) is replaced by conditioning on \( X \in N(x) \) for some appropriate neighborhood \( N(x) \).
Features and Questions

Features

• Both linear regression and $k$-nn approximate the regression function $f(x) = E[Y|X = x]$ by averaging the data.

• Least squares assumes $f(x)$ is *globally linear* and uses global averages to approximate $f(x)$.

• $k$-nn assumes $f(x)$ is *locally constant* and uses local averages to approximate $f(x)$.

Questions

• Is either method optimal?

• If not, what is the optimal method?

• How can we generalize these methods to be more flexible about what is the form of the regression (linear regression) or what counts as a neighbor ($k$-nn)?