Statistical Learning and Prediction

Marjolein Fokkema

Leiden University

m.fokkema@fsw.leidenuniv.nl

Statistical Learning

- Vast set of tools for understanding data.
  - Supervised: $Y \leftarrow f(X_1, \ldots, X_p)$; predict Y on the basis of X
  - Unsupervised: $X_1, \ldots, X_p$; finding structure in X (underlying dimensions/groups)

Statistical Learning

- The linear regression model

$$
\begin{aligned}
Y &= f(X_1, \ldots, X_p) + \epsilon \\
&= \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon
\end{aligned}
$$

  can be used for explanation and/or prediction.

- Explanation: understanding how the X's are related to Y; possibly causally.

- Prediction: if we have new observations with known values of X's, what is the expected (predicted) value of Y and how accurate are these predictions?

Explanatory Regression

- Suppose we have data and obtained estimates:

$$\hat{y}_i = 2 + 0.5x_{i1} + 1.5x_{i2}$$

- Estimated coefficients indicate magnitude of the effects, standard errors indicate variability of estimates.

- Use statistical tests for deciding whether the explanatory variables really affect the response, given the other variables.

- Adequate estimation of coefficients ($\hat{\beta}$) is assumed crucial: Accurate estimates = unbiased estimates! That is:

$$\mathbb{E}[\hat{\beta}] = \beta$$

Explanatory Regression

- Unbiased estimates can be obtained with e.g., OLS:

$$\hat{\beta} = \underset{\beta}{\arg\min} \left(\sum_{i=1}^{N} (y_i - x_i^\top \beta)^2 \right)$$

- ”From a Bayesian perspective, the principle of unbiasedness is reasonable in the limit of large samples, but otherwise it is potentially misleading” (Gelman et al., 1995)

Predictive Regression

- Suppose we have data and have obtained estimates:

$$\hat{Y} = 2 + 0.5X_1 + 1.5X_2$$

- Suppose we have a new observation $x_0 = [\,2 \quad 3\,]$
- With these values we can predict Y, i.e., $2 + 0.5 \times 2 + 1.5 \times 3 = 7.5$
- We do not care to recover parameters that generated the data, but want to obtain a model that yields as accurate as possible $\hat{Y}$.
- E.g., minimize

$$\mathbb{E}[(\hat{Y} - Y)^2]$$

Exercise 1: Bias can be beneficial

- Generate some training observations and combine them into a training dataset. Use `set.seed()` to allow for later replication:

  ```
  x <- runif(50, min = -3, max = 3)
  epsilon <- rnorm(50)
  y <- 0.1*x + epsilon
  train_dat <- data.frame(x, y)
  ```

- Generate 1,000 new observations from the same distributions. Combine the observations into a test dataset.

- Generate a range of shrinkage factor values:

  ```
  s <- seq(0, 1, by = .1)
  ```

- Estimate an OLS regression on the training observations, using `lm`. Omit the intercept, by adding `0` or `-1` in the right-hand side of the model formula. Extract the estimated β coefficient using `coef`.

- Compute predictions for the test observations, but do not use function `predict`. Instead, compute predictions yourself and apply shrinkage to the OLS coefficient you just estimated: $\hat{Y} = X \cdot s \cdot \hat{\beta}$.

  Thus, generate predictions for the test observations 11 times: once for each value of the shrinkage factor s.

- For each value of s, compute MSE (mean squared prediction error) on the test observations.

- Plot the test MSE values as a function of shrinkage factor s. Is shrinkage beneficial for prediction? What is the optimal value for shrinkage factor s?

- Repeat the above experiment 100 times, and plot the average MSE (over the 100 replications) as a function of shrinkage s.

- What do you expect to happen to the *optimal* value of the shrinkage if sample size doubles (i.e., $N_{train} = 100$ instead of 50)? And if effect size doubles (i.e., the effect of x is .20 instead of .10)?

Predictive Regression

- Traditional statistical textbooks focus on obtaining *unbiased* estimates (e.g., OLS, ML).

- (Modern) statistical learning accepts biased parameter estimates as long as *variance* decreases more than *squared bias* increases.

Predictive perspective: Bias-variance trade-off

- We have a probability distribution $P*$, from which we draw a sample $\mathcal{T}$ of size N

- Let f_B be Bayes optimal f (unknown, depends on $P*$)

- Let $\bar{f}(X) = \mathbb{E}_{\mathcal{T}}[\hat{f}(X)]$

- Aim is to minimize *expected prediction error:*

$$\begin{aligned}\mathbb{E}_{\mathcal{T}}[\mathrm{EPE}(\hat{f})] = &\mathbb{E}_X[Var(Y|X)] + \\ &\mathbb{E}_X[(f_B(X) - \bar{f}(X))^2] + \\ &\mathbb{E}_{\mathcal{T}}\mathbb{E}_{X,Y}[(\hat{f}(X) - \bar{f}(X))^2]\end{aligned}$$

- EPE = irreducible error (or σ^2) + bias^2 + variance

Population: Non-linear Regression

Often we fit a linear model, assuming that the conditional means in the population lie on a straight line.

This assumption is most likely false! Why does it often work so well?

Excercise 2: Under- and overfitting with polynomial regression

- Generate a training and test set, each with $n = 50$, with a single predictor which has a non-linear effect on the response:

```
set.seed(42)
n <- 50
x <- runif(n, min = -5, max = 5)
y <- x + 8*sin(x/2) + rnorm(n)
train <- data.frame(x, y)
x <- runif(n, min = -5, max = 5)
y <- x + 8*sin(x/2) + rnorm(n)
test <- data.frame(x, y)
```

- Fit polynomial regression models to the training data of degree 1 trough 15, make predictions on the test set and compute the prediction error for each degree. (You can use `predict` for this exercise.)

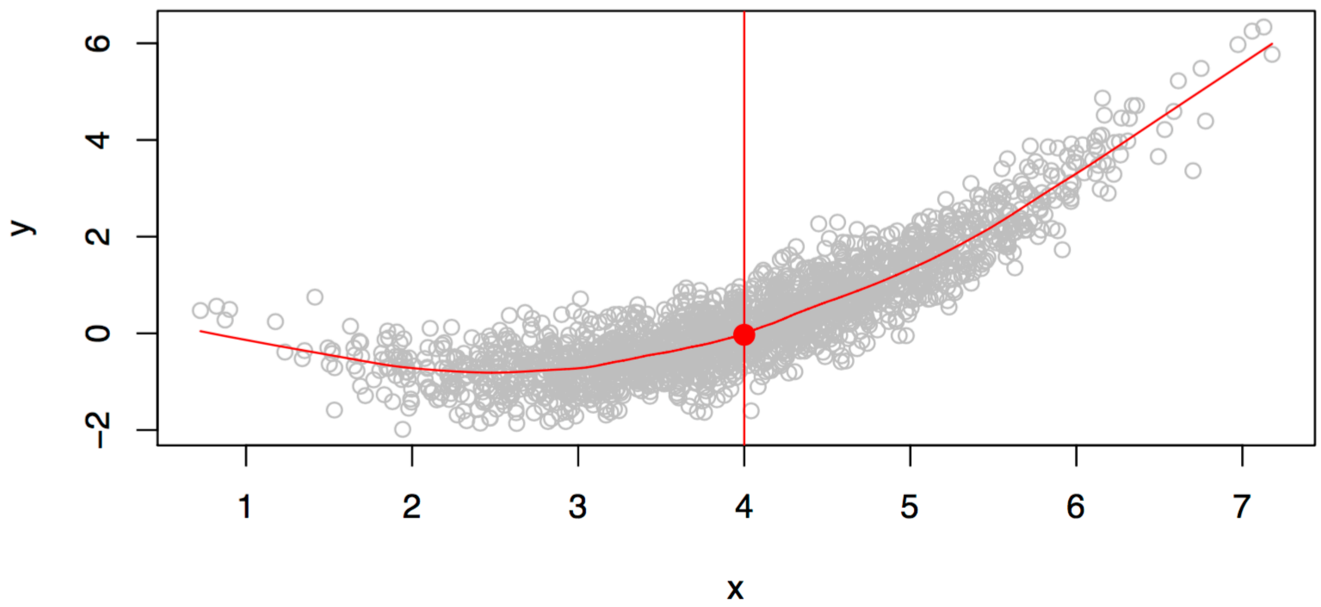
  - Hint: Use functions `lm`, `poly`, and a `for` loop. E.g.:

    ```
    lm(y ~ poly(x, degree = 3))
    ```

- Plot the test MSE as a function of the polynomial degree.

- If time permits, create a plot which shows the training observations, and fitted curves for the degree 1, 2, 3 and 15 polynomials.

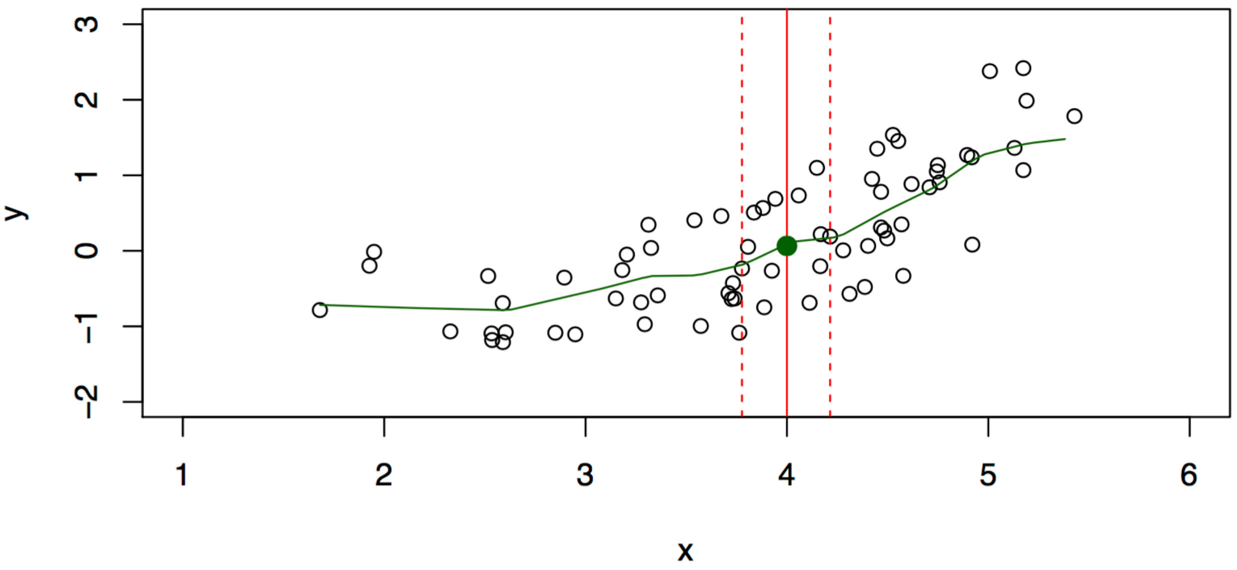
Population: Non-linear Regression



The regression line in the population combines the conditional means at each point x

Sample data

Using sample data, we want to obtain an estimate $\hat{f}(X)$ of $f(X)$.



- Due to sparsity, cannot estimate a conditional mean at all points $X = x$.

- Thus, take a small neighbourhood around $X = x$ and take neighbourhood mean as predicted value, i.e. *nearest neighbour averaging*.

- How is the size of the neighbourhood defined in kNN?
- What happens to the *bias* if k increases?
- What happens to the *variance* ik k increases?
- Does the optimal value for k depend on sample size?
- Does the optimal value for k depend on the amount of irreducible error?

Multiple Predictor Variables

With multiple predictors the observations are further spread out through the space:

- Nearest neighbours might not be near at every point
- Flexible models become very wild
- This is known as the *curse of dimensionality*
- More structure in f is needed
- *How can we impose structure?*

Exercise 3: Curse of dimensionality

- Generate a dataset with 100 observations on 10,000 predictors. Predictors may be independendent:

```
p <- 10000
N <- 100
set.seed(42)
X <- matrix(rnorm(p*N), ncol = p, nrow = N)
```

- Compute Euclidian distances between all points in the dataset. First, only use the first column of X to compute the distances (i.e., $p = 1$), then use the first two columns (i.e., $p = 2$), and so on for $p \in \{1, 2, 10, 100, 1000, 10000\}$.

- Create a histogram for each value of p.

  - Hint: use functions `dist` to compute distances, use function `hist` to create a histogram. Specify argument `xlim` for each histogram, to make sure 0 is included on the *x*-axis.

- Are the nearest neighbours near in 1-dimensional space? In 2-dimensional space? In 10-, 100-, 1000-, 10000-dimensional space?

Evaluating predictive accuracy: Numeric outcome

- Mean squared error (MSE):

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

- Mean absolute error (MAE):

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

Classification

Response variable Y may be a categorical variable with categories $\mathcal{C} = 1, \ldots, k, \ldots, K$.

Again, we want to predict response Y based on predictors X:

- Can directly construct a classifier $\hat{f}(X) = C(X)$ that assigns a predicted category from C based on X.

- Preferable, though: Construct a function $\hat{f}(X)$ that provides conditional probabilities: $\hat{p}_k(X) = Pr(Y = k|X = x)$.

  Then Bayes classifier assigns $C(X) = k$ if $\hat{p}_k(x) = \max\{\hat{p}_1(x), \ldots, \hat{p}_K(x)\}$

Evaluating predictive accuracy: Classification

- Misclassification rate:

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

- Brier score:

$$\frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

- Cross-entropy (a.k.a. deviance):

$$-\frac{1}{N}\sum_{i=1}^{N} [y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)]$$

Exercise 4: Flexibility and predictive performance

For each of parts (a) through (d), indicate whether we would generally expect the performance of a flexible statistical learning method to be better or worse than an inflexible method:

a) Sample size N is extremely large, and the number of predictors p is small.

b) The number of predictors p is extremely large, and the number of observations N is small.

c) The relationship between the predictors and response is highly non-linear.

d) The variance of the error terms, $\sigma^2 = Var(\epsilon)$, is extremely high.