CPSC 8420 Advanced Machine Learning
Week 2: Introduction to Statistical Learning

Dr. Kai Liu

August 25, 2020

Overview of Statistical learning

Statistical learning refers to a vast set of tools for understanding data. These tools can be classified as supervised or unsupervised.

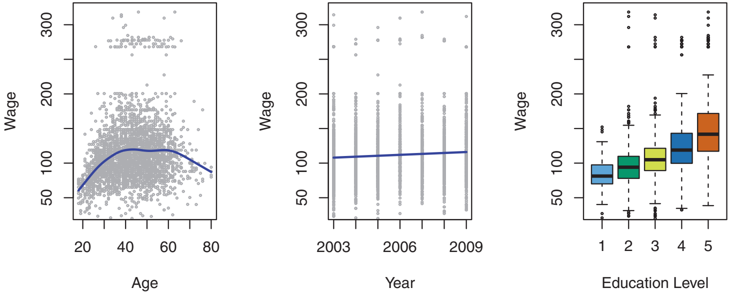


FIGURE 1.1. `Wage` *data, which contains income survey information for males from the central Atlantic region of the United States.* Left: `wage` *as a function of* `age`*. On average,* `wage` *increases with* `age` *until about* 60 *years of age, at which point it begins to decline.* Center: `wage` *as a function of* `year`*. There is a slow but steady increase of approximately* \$10,000 *in the average* `wage` *between* 2003 *and* 2009*.* Right: *Boxplots displaying* `wage` *as a function of* `education`*, with* 1 *indicating the lowest level (no high school diploma) and* 5 *the highest level (an advanced graduate degree). On average,* `wage` *increases with the level of education.*

Learning Outcomes

Our goal for this lecture is to understand:

- the difference between supervised and unsupervised learning,
- the difference between prediction and inference,
- the basic steps involved in training a model on data for a supervised learning problem,
- the fundamental trade-off between flexible and less-flexible models,
- the bias-variance trade-off.

Supervised vs Unsupervised Learning

Supervised Learning

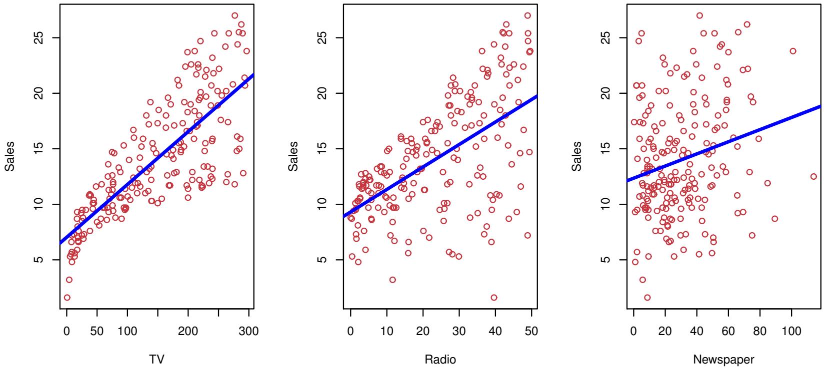
Example: Spending on advertising to increase sales

- product sales (number of units) for 200 markets
- advertising budget for TV, radio, and newspaper in each market

Questions:

- How many units will be sold in a new market for a given advertising budget?
- How much will sales increase if more money is spend on advertising?
- Which type of advertising is most successful in increasing sales, TV, radio, or newspapers?

Advertising data



We can predict sales using:

$$sales = f(tv, radio, newspaper) + noise \tag{1}$$

Notation

- *sales* is a response or target that we wish to predict. We generically denote the response measure as Y.
- *tv*, *radio*, and *newspaper* are features, or inputs, or predictors. We generally denote input as $X = (X_1, X_2, \ldots, X_p)$.
- We can now write any input-output model with a quantitative response as

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

  where f represents the systematic information that the predictors X provide about the response Y, and ϵ captures measurement errors.
- **Statistical learning refers to a set of approaches for estimating f**

Unsupervised Learning

In supervised learning, we have both predictors and outcome measures for each observation in the training data: $(X_1, Y_1), \ldots, (X_n, Y_n)$. While in unsupervised learning, we have predictors X_i but no associated responses Y_i. *e.g.* K-means

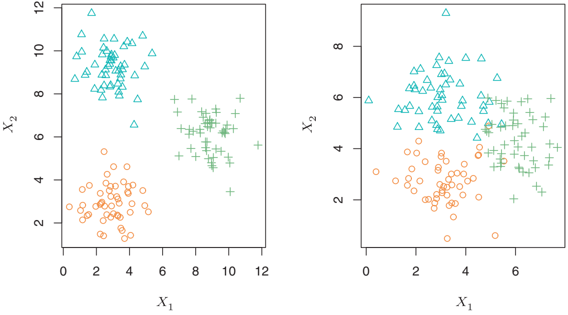


FIGURE 2.8. *A clustering data set involving three groups. Each group is shown using a different colored symbol.* Left: *The three groups are well-separated. In this setting, a clustering approach should successfully identify the three groups.* Right: *There is some overlap among the groups. Now the clustering task is more challenging.*

Why and how do we estimate f?

Why estimate f?

- **Prediction**: $\hat{Y} = \hat{f}(X)$, where $\hat{f}$ is a black box
- **Inference**: How Y is changing as a function of X
- Depending on whether the ultimate goal is prediction, inference or a mix, we may deploy different methods for estimating f.
- Depending on the ultimate goal you may or may not care about evaluating the causal relationship between Y and X.

Prediction

Examples:

- How many units will be sold in a new market for a given advertising budget?
- How much will sales increase if more money is spend on advertising?
- A college graduate with 16 years of education and 0 years seniority is starting her first job. What will her income be?

Prediction

Supervised learning for prediction is a two-step process:

- Use data $(X_1, Y_1), \ldots, (X_n, Y_n)$ to estimate $\hat{f}$
- Use $\hat{f}$ to estimate $\hat{Y}$ for a new X:

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

Prediction

Supervised learning for prediction is a two-step process:

- Use data $(X_1, Y_1), \ldots, (X_n, Y_n)$ to estimate $\hat{f}$
- Use $\hat{f}$ to estimate $\hat{Y}$ for a new X:

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

How accurately can we predict $\hat{Y}$?

Prediction

Supervised learning for prediction is a two-step process:

- Use data $(X_1, Y_1), \ldots, (X_n, Y_n)$ to estimate $\hat{f}$
- Use $\hat{f}$ to estimate $\hat{Y}$ for a new X:

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

How accurately can we predict $\hat{Y}$?
Bias: $Y - \hat{Y} = f(X) + \epsilon - \hat{f}(X)$

$$\begin{aligned} E(Y - \hat{Y})^2 &= E[f(X) + \epsilon - \hat{f}(X)]^2 \\ &= \underbrace{[f(X) - \hat{f}(X)]^2}_{\text{Reducible}} + \underbrace{\mathrm{Var}(\epsilon)}_{\text{Irreducible}} \ , \end{aligned}$$

Inference

Examples:

- Which media generate the biggest boost in sales?
- How much increase in sales is associated with a given increase in TV advertising?
- Is education or seniority more important for income?

In inference, we want to learn about relationships between predictors and Y:

- Which predictors are associated with the response?
- What is the relationship between the response and each predictor?
- Can the relationship between Y and each predictor be adequately summarized using a linear equation, or is the relationship more complicated?

How do we estimate $\hat{f}$

How do we estimate f from the training data $(X_1, Y_1), \ldots, (X_n, Y_n)$?

1. Select a statistical model / algorithm.
2. Estimate the parameters of the model from the training data.

In general, two types of statistical models:

- **Parametric**: assumes the data-generating process follows a probability distribution with a fixed set of parameters
- **Non-parametric**: does not assume (or makes fewer assumptions) about the shape or parameters of the population distribution that generated the data

Parametric methods

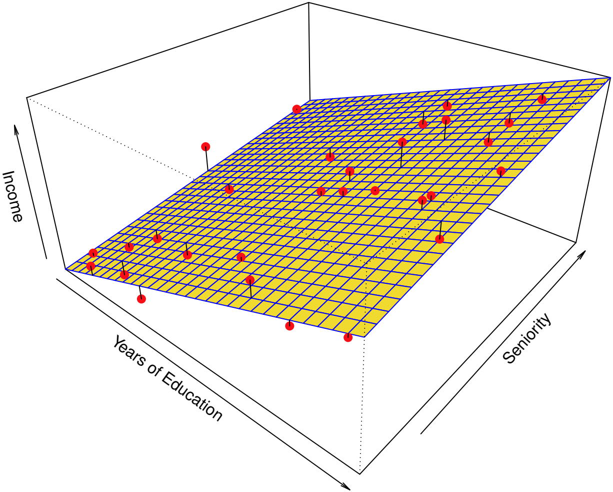
Parametric methods involve a two-step model-based approach:

1. Assume a functional form of f, e.g. that f is linear in X:

$$f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

2. Select a method to *fit* the model (e.g. *ordinary least squares (OLS)*) to estimate values for the parameters $\beta_0, \beta_1, \ldots, \beta_p$

Parametric methods

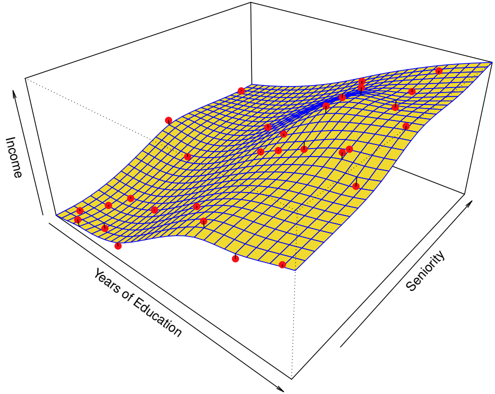


Non-parametric methods

- Not explicit (or fewer) assumptions about the functional form of f
- Goal is to estimate f such that f is as close to the data points as possible without overfitting

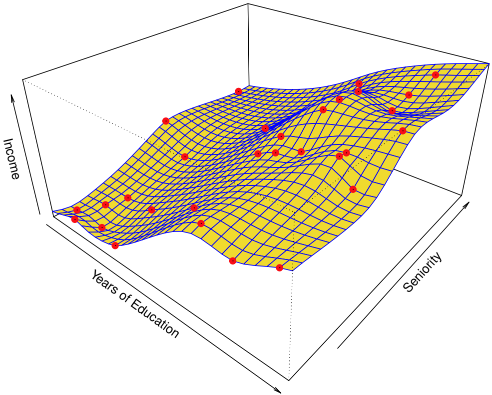
Non-parametric methods

Example: a smooth thin-plate spline fit



Non-parametric methods

Example: a rough thin-plate spline fit

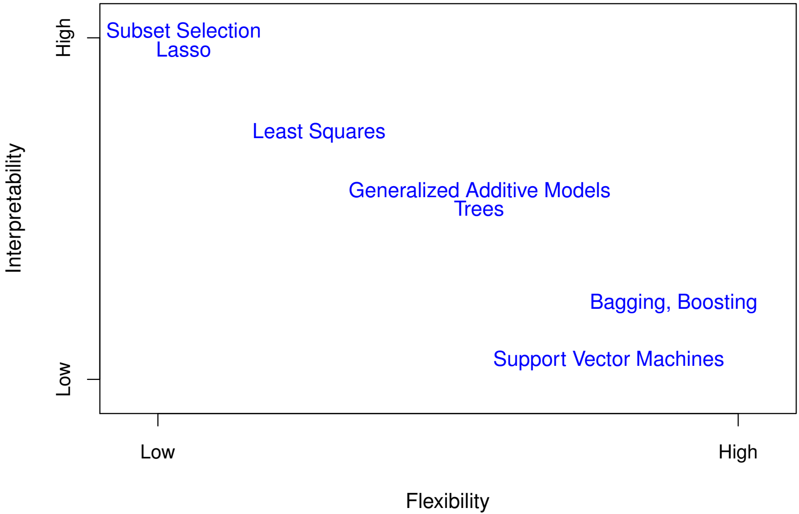


- Here the fitted model makes no errors on the training data.
- Also known as overfitting.

Trade-offs

- Prediction accuracy versus interpretability.
  - Linear models are easy to interpret; thin-plate splines are not.
- Good fit versus over-fit or under-fit.
  - How do we know when the fit is just right?
- Parsimony versus black-box.
  - We often (especially for inference) prefer a simpler model involving fewer predictors over a black-box involving many predictors.

Trade-offs


High
Low
Interpretability
Subset Selection
Lasso
Least Squares
Generalized Additive Models
Trees
Bagging, Boosting
Support Vector Machines
Low
High
Flexibility

Assessing Model Accuracy

Assessing Model Accuracy

- Suppose we train a model $\hat{f}(x)$ on some *training data* $Tr = \{x_i, y_i\}_1^N$
- We can compute the average squared prediction error over Tr:

$$MSE_{Tr} = Ave_{i \in Tr}[y_i - \hat{f}(X_i)]^2$$

But MSE_{Tr} may be biased towards models that overfit the data.

- Instead, we should compute the prediction error on a new *test data* $Te = \{x_i, y_i\}_1^M$:

$$MSE_{Te} = Ave_{i \in Te}[y_i - \hat{f}(X_i)]^2$$

Bias-Variance Trade-Off

- Suppose we have fit a model $f(x)$ to some training data Tr, and let (x_0, y_0) be a test observation drawn from the population.
- If the true model is $Y = f(X) + \epsilon$, then it can be shown that the expected test MSE for x_0 can be decomposed into the sum of three quantities:

$$E(y_0 - \hat{f}(x_0))^2 = Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2 + Var(\epsilon)$$

  where $Bias(\hat{f}(x_0)) = E[\hat{f}(x_0)] - f(x_o)$.

- Trade-off: Typically, as the flexibility of $\hat{f}$ increases, bias decreases, but variance increases.
- **The challenge lies in finding a method for which both the variance and the squared bias are low.**

KNN

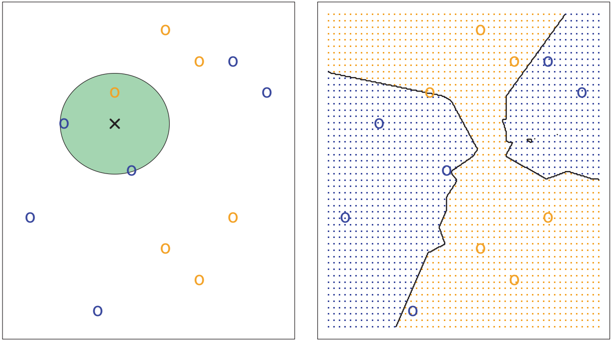


FIGURE 2.14. *The KNN approach, using $K = 3$, is illustrated in a simple situation with six blue observations and six orange observations.* Left: *a test observation at which a predicted class label is desired is shown as a black cross. The three closest points to the test observation are identified, and it is predicted that the test observation belongs to the most commonly-occurring class, in this case blue.* Right: *The KNN decision boundary for this example is shown in black. The blue grid indicates the region in which a test observation will be assigned to the blue class, and the orange grid indicates the region in which it will be assigned to the orange class.*

Trade-off

$$
\begin{aligned}
\mathrm{EPE}_k(x_0) &= \mathrm{E}[(Y - \hat{f}_k(x_0))^2 | X = x_0] \\
&= \sigma^2 + [\mathrm{Bias}^2(\hat{f}_k(x_0)) + \mathrm{Var}_{\mathcal{T}}(\hat{f}_k(x_0))] \\
&= \sigma^2 + \left[f(x_0) - \frac{1}{k} \sum_{\ell=1}^{k} f(x_{(\ell)}) \right]^2 + \frac{\sigma^2}{k}.
\end{aligned}
$$

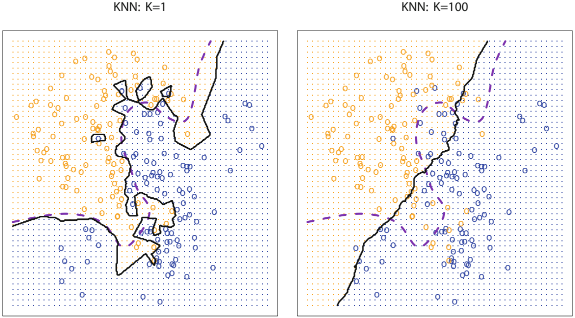


FIGURE 2.16. *A comparison of the KNN decision boundaries (solid black curves) obtained using $K = 1$ and $K = 100$ on the data from Figure 2.13. With $K = 1$, the decision boundary is overly flexible, while with $K = 100$ it is not sufficiently flexible. The Bayes decision boundary is shown as a purple dashed line.*

Balance

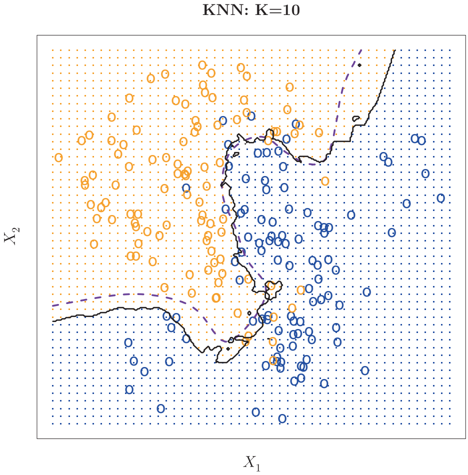


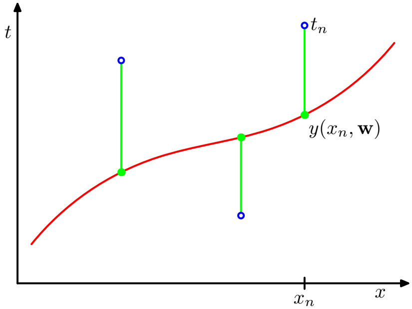
FIGURE 2.15. *The black curve indicates the KNN decision boundary on the data from Figure 2.13, using* $K = 10$*. The Bayes decision boundary is shown as a purple dashed line. The KNN and Bayes decision boundaries are very similar.*

Overcoming Overfitting

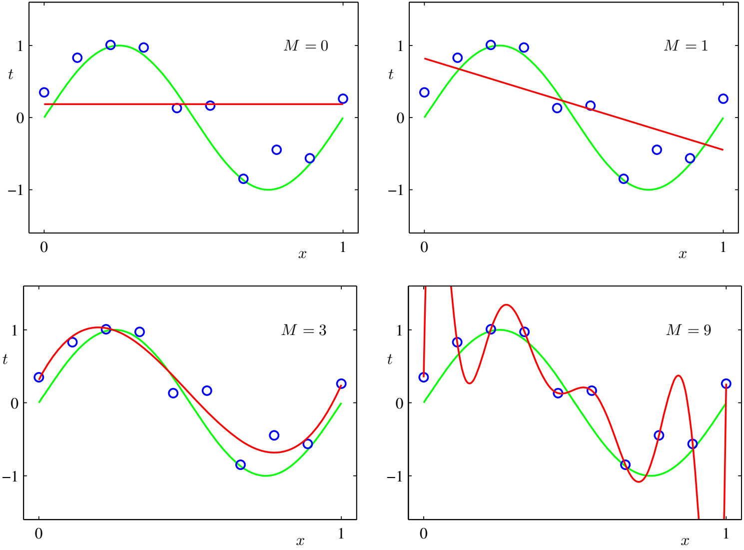
Exploring Overfitting

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2$$



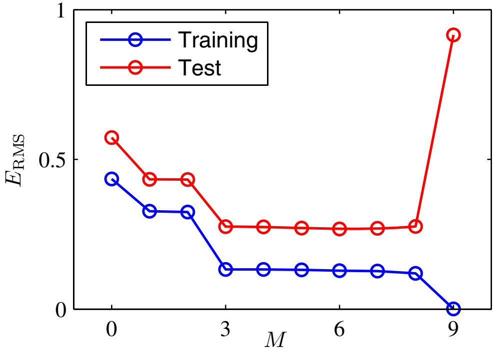
Exploring Overfitting



Exploring Overfitting

$$E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^\star)/N}$$

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2$$



Exploring Overfitting

	$M = 0$	$M = 1$	$M = 6$	$M = 9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Exploring Overfitting

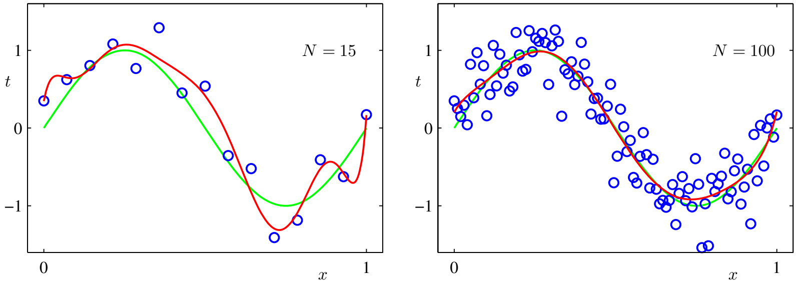
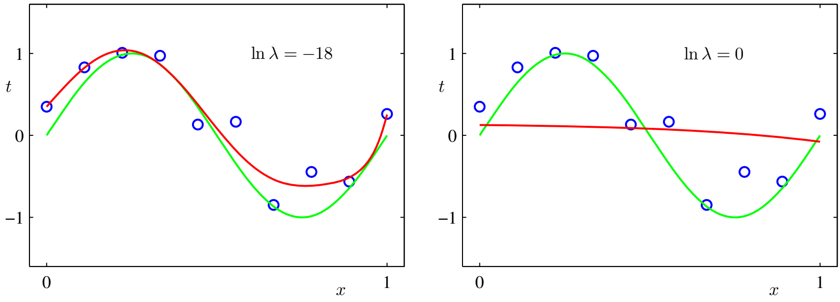


Figure 1.6 Plots of the solutions obtained by minimizing the sum-of-squares error function using the $M = 9$ polynomial for $N = 15$ data points (left plot) and $N = 100$ data points (right plot). We see that increasing the size of the data set reduces the over-fitting problem.

Exploring Overfitting

$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$



Exploring Overfitting

	$\ln\lambda = -\infty$	$\ln\lambda = -18$	$\ln\lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01

Exploring Overfitting