RECSM Summer School: Machine Learning for Social Sciences

Session 1.3: Supervised Learning and Model Accuracy

Reto Wüest

Department of Political Science and International Relations
University of Geneva

Supervised Learning

Supervised Learning

Statistical Decision Theory

Statistical Decision Theory

- Let $X \in \mathbb{R}^p$ be a vector of input variables and $Y \in \mathbb{R}$ an output variable, with joint distribution $\Pr(X, Y)$.
- Our goal is to find a function $f(X)$ for predicting Y given values of X.
- We need a loss function $L(Y, f(X))$ that penalizes errors in prediction.
- The most common loss function is squared error loss

$$L(Y, f(X)) = (Y - f(X))^2. \tag{1.3.1}$$

Statistical Decision Theory

- The expected prediction error or expected test error is

$$\text{expected test error} = E(Y - f(X))^2. \tag{1.3.2}$$

- We choose f so as to minimize the expected test error.
- The solution is the conditional expectation

$$f(x) = E(Y \mid X = x). \tag{1.3.3}$$

- Hence, the best prediction of Y at point $X = x$ is the conditional expectation.
- Let's look at two simple methods that differ in how they approximate the conditional expectation.

Supervised Learning

Method I: Linear Model and Least Squares

Linear Model and Least Squares

- In linear regression, we specify a model to estimate the conditional expectation in (1.3.3)

$$f(x) = x^T\beta. \tag{1.3.4}$$

- Using the method of least squares, we choose β to minimize the residual sum of squares

$$RSS(\beta) = \sum_{i=1}^{N}(y_i - x_i^T\beta)^2. \tag{1.3.5}$$

Linear Model and Least Squares – Example

- Goal is to predict outcome variable $G \in \{\text{blue}, \text{orange}\}$ on the basis of training data on inputs $X_1 \in \mathbb{R}$ and $X_2 \in \mathbb{R}$.
- We fit a linear regression to the training data, with Y coded as 0 for blue and 1 for orange.
- Fitted values $\hat{Y}$ are converted to a fitted variable $\hat{G}$ as follows

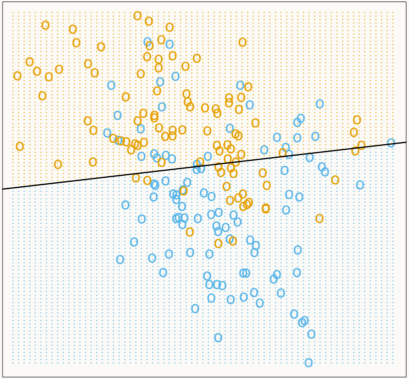
$$\hat{G} = \begin{cases} \text{orange} & \text{if } \hat{Y} > 0.5, \\ \text{blue} & \text{if } \hat{Y} \leq 0.5. \end{cases} \tag{1.3.6}$$

- In the figure below, the set of points classified as orange is $\{x \in \mathbb{R}^2 : x^T\hat{\beta} > 0.5\}$ and the set of points classified as blue is $\{x \in \mathbb{R}^2 : x^T\hat{\beta} \leq 0.5\}$. The linear decision boundary separating the two predicted classes is $\{x \in \mathbb{R}^2 : x^T\hat{\beta} = 0.5\}$.

Linear Model and Least Squares – Example

- Several training observations are misclassified on both sides of the decision boundary.

Linear Regression



(Source: Hastie et al. 2009, 13)

Supervised Learning

Method II: K-Nearest Neighbors

K-Nearest Neighbors

- K-nearest neighbors (KNN) directly estimates the conditional expectation in (1.3.3) using the training data.
- However, instead of conditioning on x, KNN uses the K observations in the training set that are closest in input space to x to form an estimate of the conditional expectation:

$$\hat{f}(x) = \frac{1}{K} \sum_{x_i \in \mathcal{N}_K(x)} y_i, \tag{1.3.7}$$

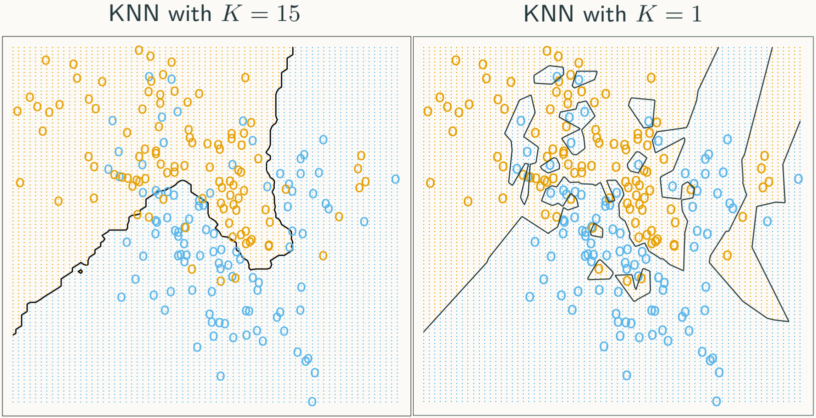
  where $\mathcal{N}_K(x)$ is the neighborhood of x defined by the K closest training observations x_i (in terms of Euclidean distance).

K-Nearest Neighbors – Example

- When KNN is applied to the above training data, $\hat{Y}$ is the proportion of orange outcomes in the neighborhood $\mathcal{N}_K(x)$.
- Creating $\hat{G}$ according to rule (1.3.6) amounts to a majority vote in the neighborhood.
- In the figures below, the decision boundaries are more irregular than the decision boundary resulting from linear regression.

K-Nearest Neighbors – Example

- Fewer (left) / none (right) training observations are misclassified than in the classification by linear regression.



KNN with $K = 15$

KNN with $K = 1$

(Source: Hastie et al. 2009, 15f.)

Supervised Learning

Linear Regression vs. K-Nearest Neighbors

Linear Regression vs. *K*-Nearest Neighbors

- Linear model assumes that $f(x)$ is well approximated by a globally linear function: its predictions are stable but possibly inaccurate (low variance and high bias).
- KNN assumes that $f(x)$ is well approximated by a locally constant function: its predictions are often accurate but can be unstable (low bias and high variance).

Linear Regression vs. *K*-Nearest Neighbors

- Should we choose the stable but biased linear model or the less biased but less stable KNN method?
- Perhaps, with a large set of training data, we can always approximate the theoretically optimal conditional expectation by KNN?
- No! If the input space is high-dimensional, then the nearest training observations need not be close to the target point (curse of dimensionality).
- KNN may be inappropriate even in low dimensions if more structured approaches can make more efficient use of the data.

Assessing Model Accuracy

Assessing Model Accuracy

- Our goal is to find a learning method $\hat{f}(X)$ to predict output Y on the basis of a set of inputs X.
- There are many methods available, so the question becomes how we should select $\hat{f}(X)$.
- Is there perhaps a "universal" method that performs well on all learning tasks?

Assessing Model Accuracy

No-Free-Lunch Theorem

There is no universal learning method that performs best on all learning tasks.

Assessing Model Accuracy

- When choosing among learning methods for a given data set, we are interested in the methods' generalization performance.
- The generalization performance of a learning method relates to its prediction accuracy on independent test data.
- Assessment of generalization performance is very important, since it guides our choice of method for a learning task.

Assessing Model Accuracy

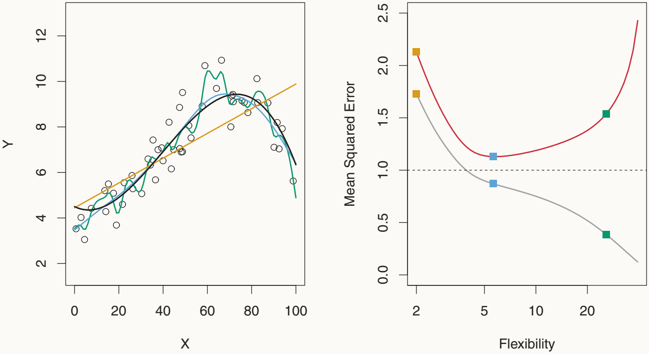
Regression

Model Accuracy in Regression Problems

- The most common performance measure is the mean squared error (MSE)
$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{f}(x_i)\right)^2, \qquad (1.3.8)$$
where $\hat{f}(x_i)$ is the prediction that $\hat{f}$ produces for the ith observation.
- The MSE in (1.3.8) is computed using the training data, so it is the training MSE.
- However, what we care about is how well the method performs on new (i.e., previously unseen) test data x_0.
- We therefore select the method that minimizes the expected test MSE
$$\text{expected test MSE} = E\left(y_0 - \hat{f}(x_0)\right)^2. \qquad (1.3.9)$$

Model Accuracy in Regression Problems

- What happens if we select the method that minimizes the training MSE in (1.3.8)?
- Danger of overfitting data: a model that is less flexible than the one we selected would have yielded a smaller test MSE.



(Left: data simulated from true f in black; orange, blue, and green curves are three estimates for f with increasing levels of flexibility. Right: training MSE in gray; test MSE in red. Source: James et al. 2013, 31)

Assessing Model Accuracy

Bias-Variance Trade-Off

Bias-Variance Trade-Off

- The U-shape in the test MSE curve is the result of two competing properties of learning methods.
- Suppose $Y = f(X) + \varepsilon$, where $E(\varepsilon) = 0$ and $Var(\varepsilon) = \sigma^2$.
- The expected test MSE of $\hat{f}(X)$ at $X = x_0$ can be decomposed into the sum of three quantities

$$\begin{aligned} \text{expected test MSE} &= E\left[(Y - \hat{f}(x_0))^2 \,\middle|\, X = x_0\right] \qquad (1.3.10) \\ &= \left[E\left(\hat{f}(x_0)\right) - f(x_0)\right]^2 \\ &\quad + E\left[\hat{f}(x_0) - E\left(\hat{f}(x_0)\right)\right]^2 + \sigma^2 \\ &= \text{Bias}^2\left(\hat{f}(x_0)\right) + Var\left(\hat{f}(x_0)\right) + \sigma^2, \end{aligned}$$

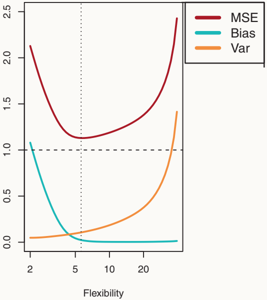
where σ^2 is the variance of the target around its true mean $f(x_0)$ (irreducible error).

Bias-Variance Trade-Off

- To minimize the expected test MSE, we need to select a method that simultaneously achieves low bias and low variance.
- **Bias:** The error that we introduce by approximating the true f by the estimate $\hat{f}$.
- **Variance:** Different training data sets result in a different $\hat{f}$. The variance refers to the amount by which $\hat{f}$ would change if we estimated it using a different training data set.

Bias-Variance Trade-Off

- More flexible methods have higher variance, while less flexible methods have higher bias. This is the bias-variance trade-off.



(Source: James et al. 2013, 36)

- In practice f is unobserved, making it impossible to explicitly compute the bias, variance, and test MSE for a method.
- We need to estimate the expected test MSE based on the available data (e.g., using cross-validation).

Assessing Model Accuracy

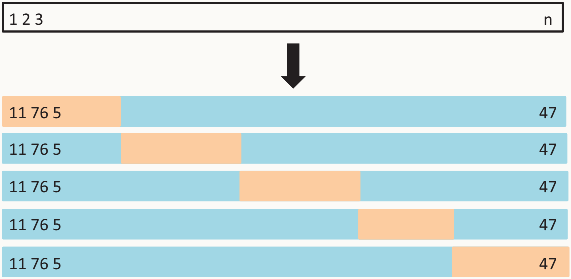
Cross-Validation

Cross-Validation

- Cross-validation (CV) is a re-sampling method that can be used to estimate the expected test error of a learning method.
- Randomly split the N training observations into $2 \leq K \leq N$ non-overlapping groups (folds) of approximately equal size.
- Use the first fold as the validation data set and the remaining folds as the training data set.
- Fit the model on the training observations.
- Use the fitted model to make predictions for the held out observations and compute the MSE.

Cross-Validation

- Repeat the procedure, each time using another fold as the validation data set. This gives K estimates of the test error, $\text{MSE}_1, \text{MSE}_2, \ldots, \text{MSE}_K$.



(Source: James et al. 2013, 181)

Cross-Validation

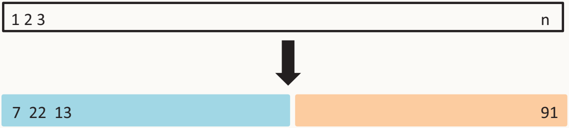
- The CV estimate for the test MSE is given by the average

$$\mathsf{CV}_{(K)} = \frac{1}{K} \sum_{k=1}^{K} \mathsf{MSE}_k. \tag{1.3.11}$$

- If $K < N$, then this procedure is called K-fold cross-validation.
- If $K = N$, then we call it leave-one-out cross-validation (LOOCV).
- Choice of K is associated with a bias-variance trade-off: LOOCV has lower bias than K-fold CV, but K-fold CV has lower variance than LOOCV.

Validation Set Approach

- In a data-rich situation, we can use the validation set approach to estimate the test error.
- Randomly split the N available observations into two groups, a training set and a validation set.
- Fit the model on the observations in the training set.
- Use the fitted model to predict the outcomes for the observations in the validation set and compute the MSE.



(Source: James et al. 2013, 181)

Assessing Model Accuracy

Classification

Model Accuracy in Classification Problems

- Suppose that we estimate f on the basis of training data $\{(x_i, y_i)\}_{i=1,\ldots,N}$, where $y_1, \ldots, y_N$ are qualitative.
- The most common approach for measuring the accuracy of $\hat{f}$ is the misclassification error

  $$\text{misclassification error} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(y_i \neq \widehat{y}_i), \quad (1.3.12)$$

  where $\widehat{y}_i$ is the predicted class label for i using $\hat{f}$ and $\mathbb{1}(y_i \neq \widehat{y}_i)$ is an indicator variable that equals 1 if $y_i \neq \widehat{y}_i$ (misclassification) and 0 if $y_i = \hat{y}_i$ (correct classification).
- The misclassification error in (1.3.12) is the training error because it is computed based on the training data.

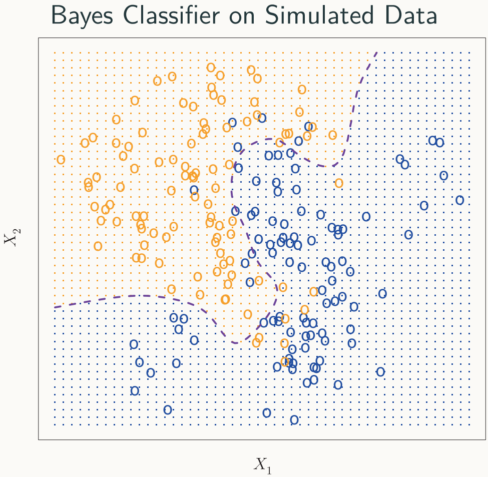
Model Accuracy in Classification Problems

- Again, however, we are more interested in selecting a method that minimizes the expected test error on new data (x_0, y_0)

$$\text{expected test error} = E\left(\mathbb{1}(y_0 \neq \hat{y}_0)\right). \qquad (1.3.13)$$

- The expected test error is minimized by the Bayes classifier, which assigns each observation to the most likely class given its predictor values, i.e., $\arg\max_{j \in \mathcal{J}} \Pr(Y = j \mid X = x_0)$.
- The Bayes classifier produces the lowest possible expected test error (called the Bayes error rate).
- The Bayes error rate is analogous to the irreducible error in the regression setting.

Model Accuracy in Classification Problems



(For each $X = x$, there is a probability that Y is orange or blue. Because the data-generating process is known, the conditional probability of each class can be calculated for each x. The orange region is the set of x for which $\Pr(Y = \text{orange} \mid X = x) > 0.5$ and the blue region is the set for which $\Pr(Y = \text{orange} \mid X = x) \leq 0.5$. The dashed line is the Bayes decision boundary. Source: James et al. 2013, 38.)

Model Accuracy in Classification Problems

- For real data, we do not know $\Pr(Y = j \mid X = x)$, so we cannot compute the Bayes classifier.
- We need to estimate $\Pr(Y \mid X)$ and then classify a given observation to the class with the highest estimated probability.
- One method to do so is KNN. Given $K \in \mathbb{Z}_{>0}$ and test observation x_0, KNN identifies the K observations in the training data closest to x_0, indicated by $\mathcal{N}_K(x_0)$, and estimates the conditional probability for each class j as the fraction of observations in $\mathcal{N}_K(x_0)$ whose output equals j

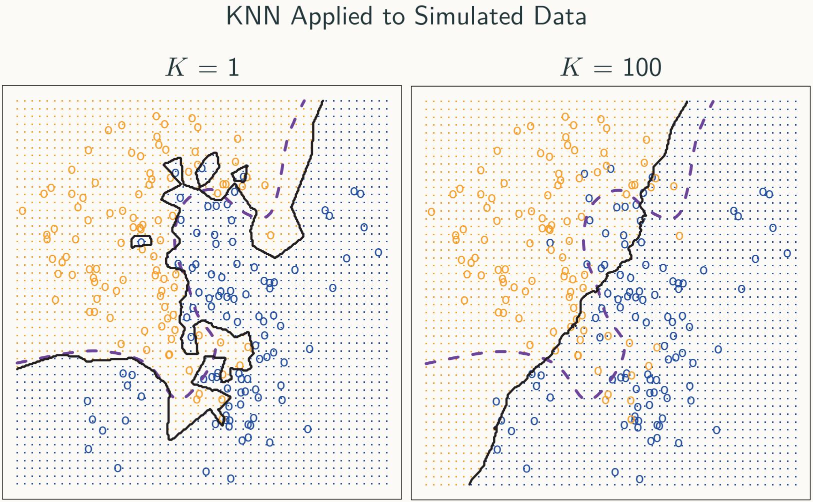
$$\widehat{\Pr}(Y = j \mid X = x_0) = \frac{1}{K} \sum_{x_i \in \mathcal{N}_K(x_0)} \mathbb{1}(y_i = j). \quad (1.3.14)$$

It then assigns x_0 to the class j with the highest probability.

Assessing Model Accuracy

Bias-Variance Trade-Off Revisited

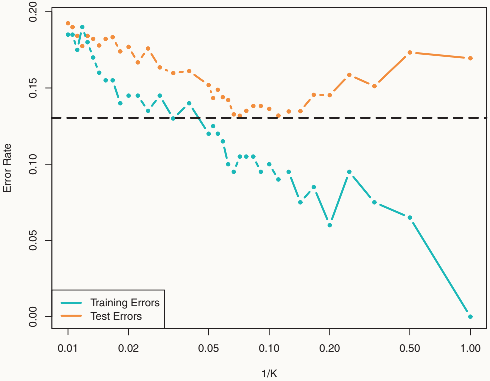
Bias-Variance Trade-Off Revisited



(KNN decision boundaries are shown as black solid lines; Bayes decision boundary is shown as a dashed line. Source: James et al. 2013, 41)

Bias-Variance Trade-Off Revisited

As $1/K$ increases, KNN becomes more flexible. As flexibility increases, the training error consistently declines and the test error exhibits the characteristic U-shape.



(Error rates as a function of flexibility $(1/K)$. Bayes error rate is indicated by a dashed line. Source: James et al. 2013, 42)

Assessing Model Accuracy

Cross-Validation Revisited

Cross-Validation Revisited

- As for regression problems, the level of flexibility is critical to the performance of a classification method.
- We can again use cross-validation to choose the optimal level of flexibility.
- However, instead of using MSE to quantify test error, we now use the number of misclassified observations.
- In the classification setting, the CV estimate for the expected test error is

$$\mathsf{CV}_{(K)} = \frac{1}{K}\sum_{k=1}^{K} \mathsf{Err}_k, \tag{1.3.15}$$

where $\mathsf{Err}_k = \frac{1}{N_k}\sum_{i=1}^{N_k} \mathbb{1}(y_i \neq \hat{y}_i)$ and N_k is the number of observations in the kth validation set.