RECSM Summer School: Machine Learning for Social Sciences

Session 1.2: General Introduction

Reto Wüest

Department of Political Science and International Relations
University of Geneva

Outline

What is Machine Learning?

Learning

The process of converting experience into knowledge.

Machine Learning

Machine learning is automated learning. We program computers so that they can learn from input available to them.

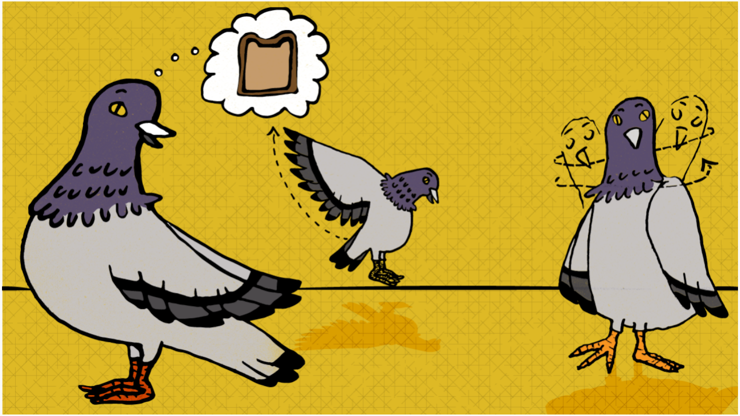
- The input to a learning algorithm is training data, representing experience.
- The output of a learning algorithm is knowledge, which we can use to perform some task (e.g., prediction, pattern detection).
- A successful learning algorithm should be able to generalize (inductive reasoning).

(Shalev-Shwartz and Ben-David 2014, 19ff.)

Learning Example I: Bait Shyness

(Image: 123rf.com)

Learning Example II: Pigeon Superstition



(Image: vocativ.com)

What Distinguishes Successful from Unsuccessful Learning?

- Incorporation of prior knowledge that biases the learning mechanism (inductive bias).
- The stronger the prior knowledge (or prior assumptions), the easier the learning from further examples.
- The stronger the prior knowledge (or prior assumptions), the less flexible the learning.
- We will come back to these issues in our discussion of the bias-variance trade-off.

When Do We Need Machine Learning?

When do we rely on machine learning rather than directly program computers to carry out the task at hand?

- **Complex tasks:** Tasks that we do not understand well enough to extract a well-defined program from our expertise (e.g., analysis of large and complex data, driving).
- **Tasks that change over time:** Machine learning tools are, by nature, adaptive to the changes in the environment they interact with (e.g., spam detection, speech recognition).

Types of Machine Learning

Supervised Learning

- Data: for every observation $i = 1, \ldots, n$, we observe a vector of inputs $\mathbf{x}_i$ and a outcome y_i.
- Goal: fit a model that relates outcome y_i to $\mathbf{x}_i$ in order to accurately predict the outcome for future observations.
- If Y is quantitative, then this problem is a regression problem; if Y is categorical, then it is a classification problem.

Unsupervised Learning

- Data: for every observation $i = 1, \ldots, n$, we observe a vector of inputs $\mathbf{x}_i$ but no associated outcome y_i.
- Goal: learning about relationships between the inputs or between the observations.

Supervised Learning

Statistical Decision Theory

- Let $X \in \mathbb{R}^p$ be a vector of input variables and $Y \in \mathbb{R}$ an outcome 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

$$L(Y, f(X)) = (Y - f(X))^2. \tag{1}$$

Statistical Decision Theory

- The expected (squared) prediction error is

$$EPE(f) = E(Y - f(X))^2. \tag{2}$$

- We choose f so as to minimize the EPE.
- The solution is the conditional expectation

$$f(x) = E(Y \mid X = x). \tag{3}$$

- Hence, the best prediction of Y at point $X = x$ is the conditional mean.

Linear Model and Least Squares

- In linear regression, we specify a model to estimate the conditional expectation in (3)

$$\hat{f}(x) = x^T \hat{\beta}. \tag{4}$$

- Using the method of least squares, we choose $\hat{\beta}$ to minimize the residual sum of squares

$$RSS(\beta) = \sum_{i=1}^{N} (y_i - x_i^T \beta)^2. \tag{5}$$

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 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

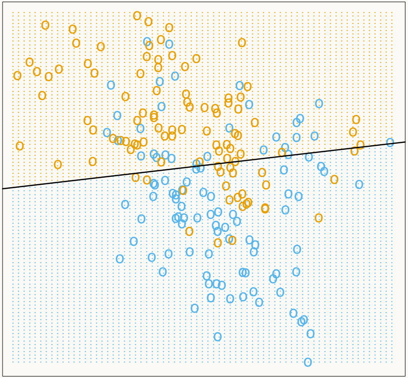
$$\hat{G} = \begin{cases} \text{orange} & \text{if } \hat{Y} > 0.5, \\ \text{blue} & \text{if } \hat{Y} \leq 0.5. \end{cases} \tag{6}$$

- 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)

k-Nearest Neighbors

- k-nearest neighbors directly estimates the conditional expectation in (3) using the training data.
- However, instead of conditioning on x, k-nearest neighbors uses the k observations in the training set that are closest in input space to x to form an estimate of the conditional expectation:

$$\hat{f}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i, \tag{7}$$

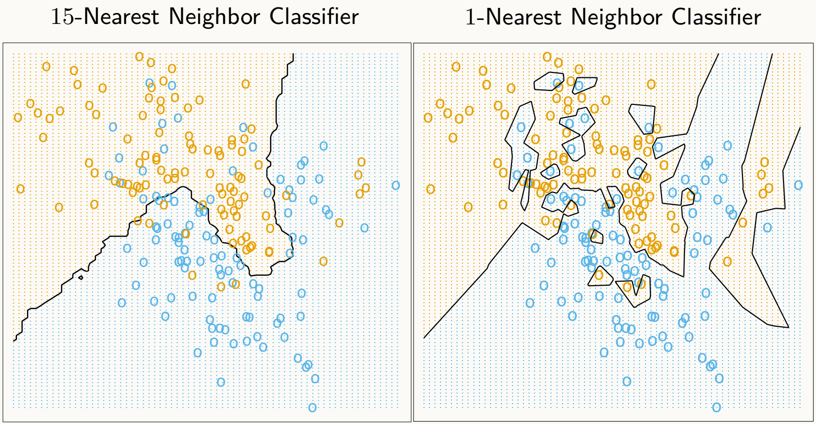
where $N_k(x)$ is the neighborhood of x defined by the k closest training observations x_i.

k-Nearest Neighbors: Example

- When nearest neighbors is applied to the above training data, $\hat{Y}$ is the proportion of orange outcomes in the neighborhood $N_k(x)$.
- Creating $\hat{G}$ according to rule (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

- Far fewer training observations are misclassified than in the classification by linear regression.



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

Linear Regression Versus 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).
- k-nearest neighbors assumes that $f(x)$ is well approximated by a locally constant function: its predictions are often accurate but can be unstable (high variance and low bias).

Linear Regression Versus k-Nearest Neighbors

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