CSCI 0410: AI Foundations **Spring 2026**
Professor Serena Booth[1] **Supervised Learning**

All models are wrong; some are useful.

– George Box

Intro to ML: Supervised Learning

1 What is Learning?

Learning is the process of associating features with a response variable. In machine learning, we develop models that can learn from data and make predictions or decisions based on new input.

2 Types of Learning Tasks

There are four main types of learning tasks:

1. **Supervised Learning**
2. **Unsupervised Learning**
3. **Semi-supervised Learning**
4. **Reinforcement Learning**

3 Supervised Learning

In supervised learning, we have labeled data, meaning each example in our dataset has an associated correct output.

3.1 Mathematical Representation

Let:

- $X \in \mathbb{R}^{n \times d}$ be the input data or feature matrix,
- where n is the number of examples and d is the number of features.
- Y represents the labels (correct predictions or outputs).

Example: Toothache Prediction

[1]This notes were compiled in conjunction with with Professors Eric Ewing and Amy Greenwald.

- X: Symptoms and observations (e.g., pain, sensitivity, etc.)
- Y: Diagnosis (e.g., having a cavity)

Goal: We aim to learn a function f such that:

$$\hat{y} = f(x) \tag{1}$$

This function, also called a **model**, should generalize well to unseen data.

4 Example: Iris Dataset

The iris dataset contains three classes:

1. Iris Setosa
2. Iris Versicolor
3. Iris Virginica

4.1 Data Description

- 50 observations ($n = 50$)
- 4 measured variables per observation ($d = 4$)
- X is a 50×4 matrix
- Goal: Predict the type of flower

The need for a model arises because we want to classify flowers we haven't seen before. This task is called **classification**.

5 Classification

- The response variable y belongs to a discrete set of classes:

$$y \in \{\text{setosa, versicolor, virginica}\} \tag{2}$$

6 Regression

- When y is a continuous value, we aim to predict:

$$y \in \mathbb{R} \tag{3}$$

7 K-Nearest Neighbors (KNN)

KNN is a simple and intuitive classification algorithm that classifies a query point based on the labels of its closest neighbors.

7.1 Steps in KNN

1. Find the k nearest data points to a given query point.
2. Determine the class of the k nearest data points.
3. Use a majority vote to determine the class of the query point.
    - If $k = 3$, and 2 neighbors belong to one class while 1 belongs to another, the majority class is assigned.

The output label you predict depends on what K actually is

Why have we only used odd numbers ? To account for ties in the majority vote

8 Decision Boundaries

Once we visualize labeled data and choose k, we observe a **decision boundary**. A decision boundary is an imaginary line or surface that separates different classes in a classification problem.

8.1 Effect of Increasing k

- Small k: The decision boundary is highly complex and sensitive to data points.
- Large k: The decision boundary becomes straighter and smoother, approaching a linear separation.

How do we reason about the above two trade-offs?

9 Bias-Variance Tradeoff

What is the best decision boundary? What does it mean to have a good model? What are some properties we'd like our model to have?

The choice of k involves a tradeoff:

- **Small** k:
    - High variance: The model is overly sensitive to small fluctuations in the dataset.
    - Low bias: The decision boundary captures noise rather than general patterns.
    - Overfitting: The model learns details specific to the training set rather than generalizing well.
- **Large** k:
    - High bias: The model is overly simplistic and does not capture enough details.
    - Low variance: The decision boundary is stable and generalizes well to new data.
    - Underfitting: The model fails to capture meaningful trends in the data.

10 Model Selection

Now we want to find a model $f(x)$ by electing an optimal k that balances bias and variance. (In this case, our model is KNN).

Given some class of K models, how do we choose the model that best classifies the irises?

How can we see how well our model is performing? We split our dataset into a train and test set to evaluate model performance

$$X = \{X_{train}, X_{test}\}$$

:

- Training set x_{train}, y_{train}: Used to fit the model.
- Test set x_{test}, y_{test}: Used to evaluate how well the model generalizes.
- A typical split is 80% training and 20% test data.

** **NOTE:** We have the actual labels of the test set (y_{test}). That is what makes this learning "supervised." Although the challenge of ML is that in the real world, we don't have the labels!

11 Model Evaluation

When we look out our decision boundary visualization, we see that there are some red data points in the blue region of the boundary. These were incorrectly classified. How can we evaluate this. We evaluate performance using:

- **Accuracy:** $\frac{\text{correct predictions}}{\text{total predictions}}$
- We'll be optimizing for some parameter (in this case, K) to maximize our test accuracy

Take the model with the highest test accuracy.

12 Distance Metrics for KNN

How do we measure "nearness" between data points?

- **Euclidean Distance:** Standard geometric distance in d-dimensional space.

$$d(x, x') = \sqrt{\sum_{i=1}^{d} (x_i - x'_i)^2}$$

- **Manhattan Distance:** Sum of absolute differences across dimensions.

$$d(x, x') = \sum_{i=1}^{d} |x_i - x'_i|$$

- **Cosine Distance:** Measures angle between two vectors.

$$d(x, x') = 1 - \frac{x \cdot x'}{\|x\|\|x'\|}$$