CAI 4104/6108 – Machine Learning Engineering:

Introduction

Prof. Vincent Bindschaedler

Spring 2024

Administrivia

- Setting up your environment
  - Happy to see Slack used for this
    - If you are having issues that you cannot resolve, let us know (Slack, Canvas, email, office hours)

- Office Hours
  - Mondays @ 11:30 AM (online) or by appointment

- TA Office Hours
  - Mondays and Wednesday @ 2:00 PM (online)
  - Fridays @ 4:30 PM (online)

- Background Survey (not graded)
  - Average score is about 60% so far

What is Machine Learning?

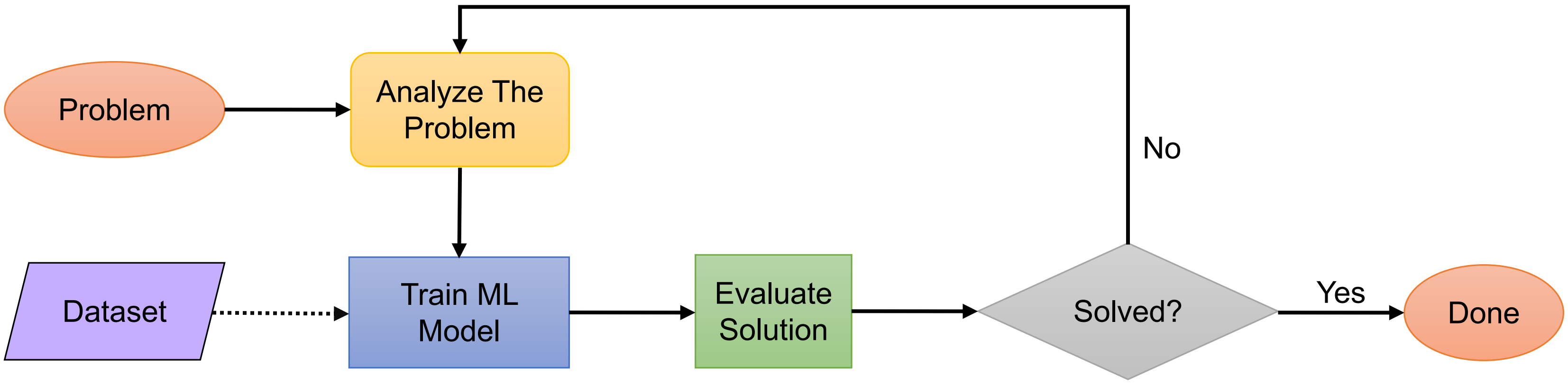
- Definitions:
  - *"Machine learning is the study of computer algorithms that improve automatically through experience."*
    - Tom Mitchell, 1997.
  - *"Machine learning is programming computers to optimize a performance criterion using example data or past experience."*
    - Ethem Alpaydin, 2004.

- What makes an algorithm a machine learning algorithm?

- In this course:
  - Machine learning is about using data to create models (broadly construed) that can make predictions (broadly construed)
    - Note: old(er) techniques like linear models, decision tables, or decision trees are still ML
  - Terminology: "machine learning" = "learning"

What is Machine Learning?

- An alternative view of what machine learning is about:
  1. Identify a problem
  2. Collect and prepare a dataset for it
  3. Use an algorithm to build a model from the dataset
  4. Use the model to solve the problem



Example: Avocado Ripeness

- Problem: how to tell when an avocado is ripe?
- Dataset?
  - Find one online (e.g., Kaggle's fruits 360 dataset)
  - Create one
    - Go to Publix, buy a bunch of avocados
    - For each avocado: slice it open and label it (as ripe or unripe)

- Features?
  - We could take a picture of each avocado and use the pixels' RGB as features, or
  - We could extract features manually. For example:
    - Color: light green, green, dark green, dark purple, black
    - Softness: firm, slightly firm, soft, mushy
    - Texture: smooth, bumpy
- Prediction task:
  - Given features of an avocado (i.e., color, softness, texture), predict ripe (1) or unripe/overripe (0)

Example: Avocado Ripeness

- Prediction task:
  - Given the features (i.e., color, softness, texture), predict ripe (1) or unripe/overripe (0)
- Dataset
  - Feature engineering:
    - Let's say we encode color (0, 1, 2, 3, 4), softness (0, 1, 2, 3), and texture (0, 1)
  - Suppose we have n examples, the data can be viewed as a n × 3 matrix, and a corresponding vector of n labels (0/1)
    - Call the matrix $\mathbf{X}$ and the labels vector $\mathbf{y}$
  - For example:

$$\mathbf{X} = \begin{pmatrix} 1 & 0 & 0 \\ \boxed{3 & 2 & 1} \\ 3 & 1 & 0 \\ 2 & 2 & 1 \\ 4 & 3 & 1 \end{pmatrix} \qquad \mathbf{y} = \begin{pmatrix} 0 \\ \boxed{1} \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

E.g.: the second avocado has the following features:

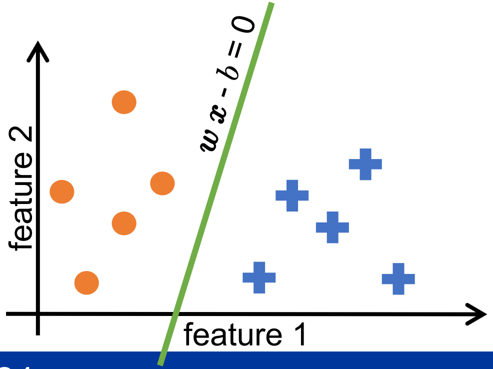
- 3 (color: dark purple),
- 2 (softness: soft),
- 1 (texture: bumpy);

It is ripe (1)

Example: Avocado Ripeness

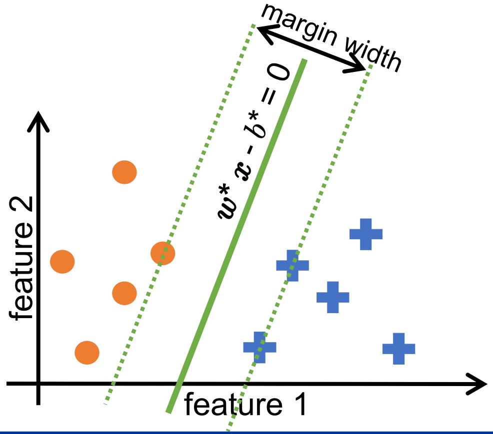
- Prediction task:
  - Given the features (i.e., color, softness, texture), predict ripe (1) or unripe/overripe (0)

- Dataset
  - Matrix $\mathbf{X}$ and the labels vector $\mathbf{y}$
- Let's use a Support Vector Machine (SVM) model:
  - We need to relabel unripe/overripe as -1, so the labels are +1 and -1
  - The SVM is represented as the hyperplane $\boldsymbol{w}\,\boldsymbol{x} - b = 0$,
    - $\boldsymbol{x}$ is a feature vector and $\boldsymbol{w}$ and b are the model's parameters (to be learned from data)
  - Define $f_{\theta}(\boldsymbol{x}) = \mathrm{sign}(\boldsymbol{w}\,\boldsymbol{x} - b)$, where $\boldsymbol{\theta} = (\boldsymbol{w}, b)$
    - If $\boldsymbol{w}\,\boldsymbol{x} - b \geq 0$, then we predict +1 (ripe)
    - Otherwise, we predict -1 (unripe/overripe)
    - Note: $\boldsymbol{w}\,\boldsymbol{x}$ is the dot-product of $\boldsymbol{w}$ and $\boldsymbol{x}$
      - $\boldsymbol{w}\,\boldsymbol{x} = w_1 x_1 + w_2 x_2 + \ldots + w_m x_m$



Example: Avocado Ripeness

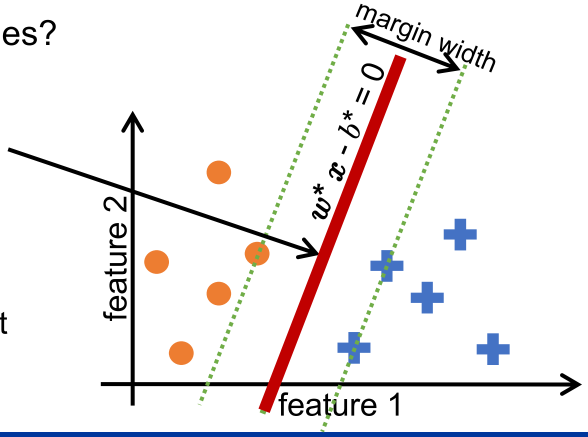
- Dataset
  - Matrix $\mathbf{X}$ and the labels vector $\mathbf{y}$
  - Let $\boldsymbol{x}_i$ be the feature vector for example i and y_i be the corresponding label
- Training the model
  - We need to learn the optimal parameter values $\boldsymbol{w}^*$, b^* given our dataset
  - We want the hyperplane that best separates positive from negative examples
    - That is: the one with the largest distance (called "margin") between the closest examples of each class
  - This is an optimization problem with constraints:
    - Why? We want: $\boldsymbol{w}\,\boldsymbol{x}_i - b \geq +1$ if y_i=+1 and $\boldsymbol{w}\,\boldsymbol{x}_i - b \leq -1$ if y_i=-1
    - Minimize $||\boldsymbol{w}||$ subject to: $y_i(\boldsymbol{w}\,\boldsymbol{x}_i - b) \geq 1$ for i=1,2,...,n
      - (This is called hard margin linear SVM in the linearly separable case.)
  - So our model is: $\theta^* = (\boldsymbol{w}^*, b^*)$
- Support Vector Machines:
  - There are different kinds of SVMs (soft vs. hard margin, kernels)
    - We will explore this later in the course



Example: Avocado Ripeness

- Dataset
  - Matrix $\mathbf{X}$ and the labels vector $\mathbf{y}$
  - Let $\boldsymbol{x}_i$ be the feature vector for example i and y_i be the corresponding label
- Training the model
  - So our model is: $\theta^* = (\boldsymbol{w}^*, b^*)$
- What if we want to classify a new example?
  - Given feature vector $\boldsymbol{x}'$ of an (unseen) avocado: $y' = \text{sign}(\boldsymbol{w}^* \boldsymbol{x}' - b^*)$
  - Question: will the model predict correctly unseen (i.e., new) examples?
    - In other words: will the model generalize?

decision boundary



- Concepts & Terminology
  - We used SVM to solve a binary classification problem
    - We learned the model by solving an optimization problem on our dataset
  - The model defines a decision boundary (hyperplane)

Avocado Ripeness: Alternative

- Dataset
  - Matrix $\mathbf{X}$ and the labels vector $\mathbf{y}$
  - Let $\boldsymbol{x}_i$ be the feature vector for example i and y_i be the corresponding label

- Do we need to train a model?
  - Given feature vector $\boldsymbol{x}$ of an avocado, we want to predict ripeness.
    - Do we necessarily need to train a model for this? **No!**
- Instance-based learning
  - What if: given the feature vector $\boldsymbol{x}$, we find the most similar example in our dataset
    - If that avocado has feature vector $\boldsymbol{x}$**'** and label y', we predict label y' (if the avocado is ripe we say ripe, otherwise we say unripe)
  - There are several ways to define similarity (e.g.: Euclidean distance, cosine similarity, etc.)
  - This strategy is called k-Nearest Neighbors classification
    - If k>1, then we take can predict using the majority label among the k most similar examples
- Note: in contrast, training an SVM model is an example of model-based learning

Evaluating Models

- To assess a model's performance, we need:
  - A metric
    - For classification, we can use accuracy (# of correctly predicted instances / total # of instances)
  - A dataset (containing features and corresponding labels)
- Should we use the training dataset to evaluate the model?
  - **No!!** We care about making accurate predictions on **new** data!
  - Our model has seen the training data; it would not be surprising if it made good predictions on it
    - *How accurate will the predictions of a kNN model with k=1 be on the training data?*
  - We need a separate test dataset

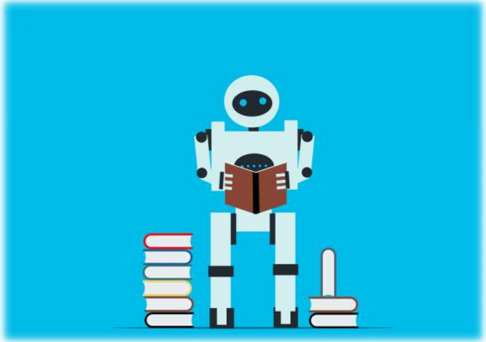
- Best practice:
  - After you have pre-processed the data, split the data into a training set and test set
    - Rule of thumb: 80% for training, 20% for testing
  - Never use the test data except at the very end of the process. **Why?**

Takeaways: Machine Learning is About:

- Solving problems using **data** by training a ***model***
  - How to solve the problem is dictated by the data, not some hardcoded algorithm

- Framed as an ***optimization problem*** with a learning objective (***loss function)***
  - So training the model means algorithmically finding the best solution (***model parameters***)

- The model needs to ***generalize*** beyond the **training data**
  - It should make **accurate predictions** on ***unseen*** **data** (e.g., test dataset)



References, Exercise & Next Time

- Book references:
  - Chapter 1 of the “Hands on ML” book (2nd ed)
  - Note: SVM is not discussed in depth until Chapter 5

- Next Meeting on Friday (1/12) — Exercise 0
  - We will dive into training ML models using scikit-learn
  - If you already **set up your environment**, feel free to follow along
    - The Jupyter notebook for ex0 can be downloaded from Canvas

source: xkcd