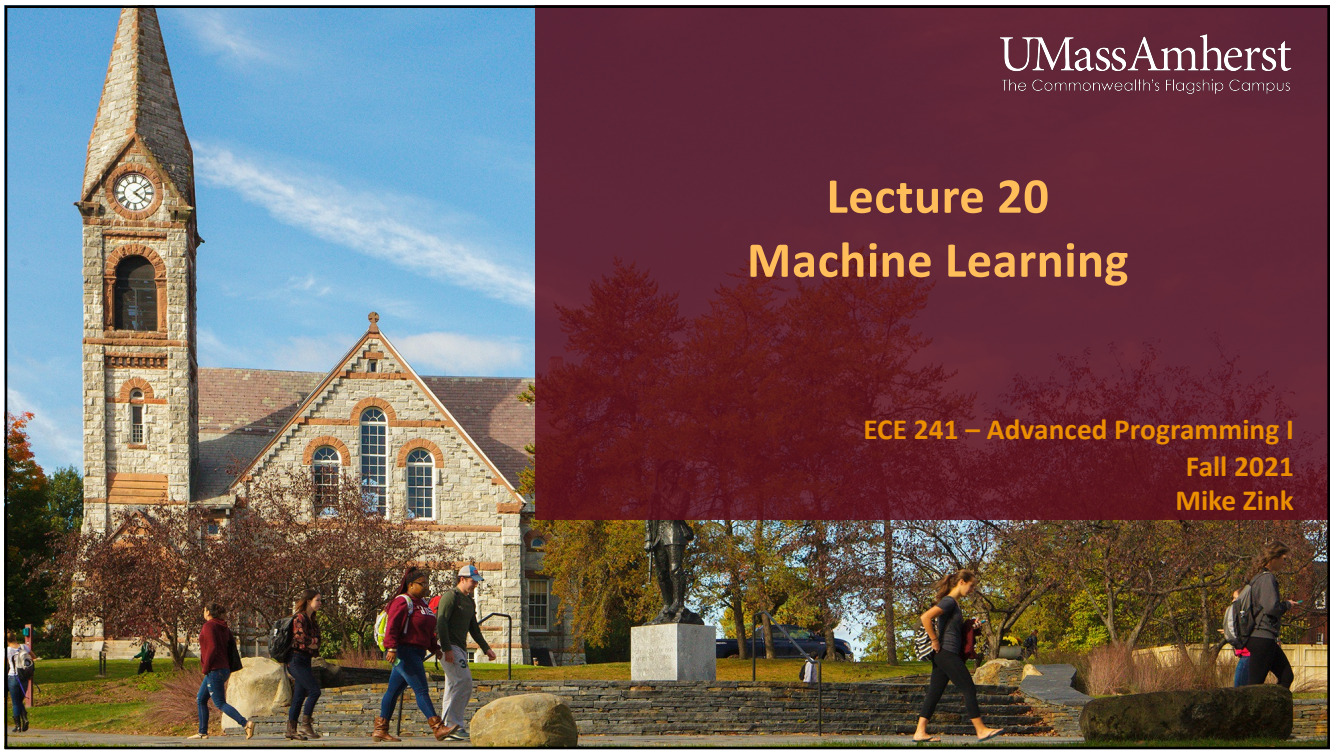


0

UMassAmherst

Introduction

- The world is filled with data! Some examples:*

Healthcare Speech Recognition
Search Engines Manufacturing Transportation
Financial Services Image Processing Retail
Spam Filters Music recommendations

- Machine Learning helps us make sense of all these data.

* See https://www.kaggle.com/datasets for a collection of datasets.

1

UMassAmherst

What is Machine Learning?

- The science and application of algorithms that help us make sense of (usually large) data
- "*Machine learning is the science of getting computers to act without being explicitly programmed*"
  by Prof. Andrew Ng
- Using data to answer questions
  Training **Prediction**

2

2

UMassAmherst

What does ML do?

- Reduce time programming
  - Custom programs with lots of rules vs. general program that you train and optimize.
    E.g., how would you write a program to check spelling errors?
- Customize and scale products
  - Expand that program with lots of rules (possibly months of programming) vs. train your program with the new data.
    E.g., how would you scale your spelling checker program to multiple languages?
- Solve problems that are perceived as unprogrammable
  - Face and speech recognition, for instance.

3

UMassAmherst

Different types of Machine Learning

- Supervised learning
  - Labeled data
  - Direct feedback
  - Predict outcome/future
- Unsupervised learning
  - No labels/targets
  - No feedback
  - Find hidden structure in data
- Reinforcement learning
  - Decision process
  - Reward system
  - Learn series of actions

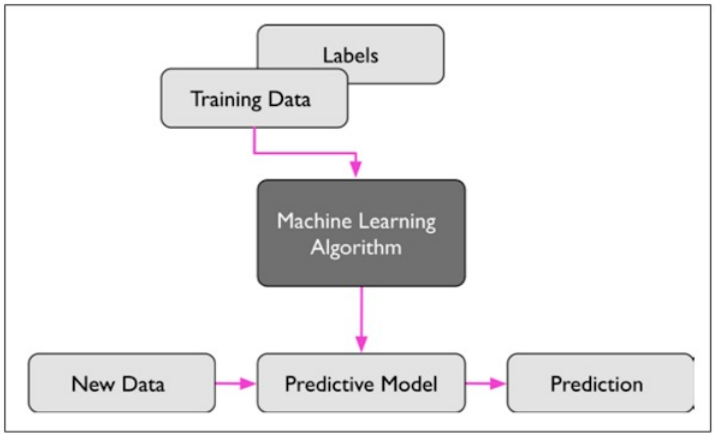
UMassAmherst

Terminology

- Label
  - The parameter that we are predicting.
  - The **y** variable in basic linear regression.
- Features
  - The input variables describing our data.
  - The $\{x_1, x_2, x_3, \ldots, x_n\}$ in basic linear regression
- Model
  - Define the relationship between label and features.

UMassAmherst

Supervised Learning



Source: Python Machine Learning, By Sebastian Raschka, Vahid Mirjalili

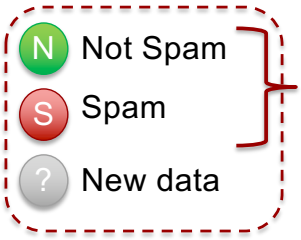
6

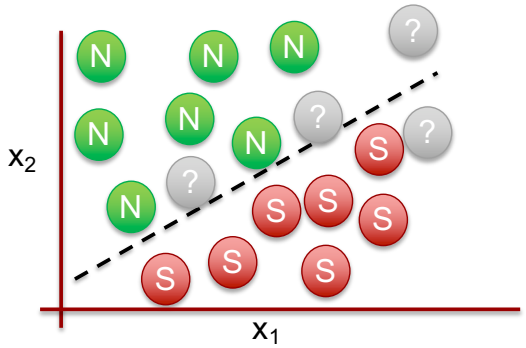
6

UMassAmherst

Classification

- The goal is to predict the category of new label instances.
- Examples:
  - Classify new emails as **Spam** or **Not Spam** (Binary)
  - Handwritten letters (Multiclass)





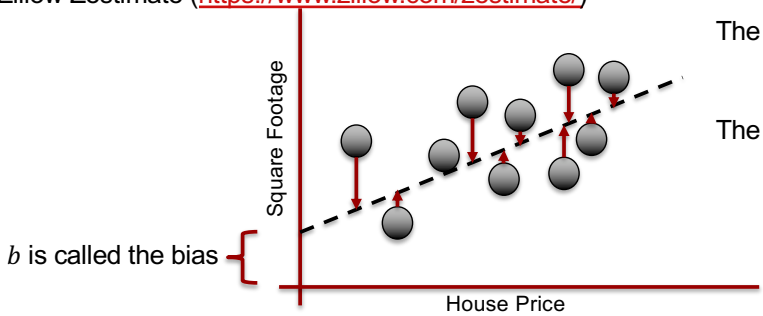
7

7

UMassAmherst

Linear Regression

- Another type of classification is regression analysis, where the goal is to predict continuous outcomes.
- For example, given the characteristics of a house, predict its price.
  See Zillow Zestimate (https://www.zillow.com/zestimate/)



The slope of the curve (weight) is $y = wx + b$

The arrow represents loss

8

8

UMassAmherst

Mean Square Error

- In reality, we are interested in the total loss, not just one individual.
- Mean square error is the square of the difference between the observed value and our model's prediction.

$$MSE = \frac{1}{N} \sum_{(x,y) \in D} (y - prediction(x))^2$$

where D is the data set.

9

9

UMassAmherst

Adding features to our model

- What if we were to add features to our house price prediction model?
  - ✓ Square footage
  - Lot size
  - Number of bedrooms
  - Number of bathrooms
  - Year built

10

10

UMassAmherst

Adding features to our model

	Sq. Ft	Lot	Beds	Bath	Year	Price
1	3,500	10,001	6	4	1890	1,299,000
2	2,295	5,227	5	3	1900	800,000
3	2,484	4,791	5	2	1910	1,250,000
…	…	…	…	…	…	…
N	1,190	4.050	3	1.5	1920	645,000

11

11

UMassAmherst

Adding features to our model

- We can represent the data as a matrix, where the i^{th} house can be written as

$$x^{(i)} = \begin{bmatrix} x_1^{(i)} & x_2^{(i)} & x_3^{(i)} & x_4^{(i)} & x_5^{(i)} \end{bmatrix}$$

- and each feature can be represented as a column vector

$$x_j = \begin{bmatrix} x_j^{(1)} \\ x_j^{(2)} \\ \vdots \\ x_j^{(N)} \end{bmatrix}$$

12

12

UMassAmherst

Adding features to our model

- And the target (price of the house) can also be represented as a column vector

$$y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}$$

- You can imagine that with millions of houses and even more features our dataset can grow large
- Thus, the computation time of our model increases

13

13

UMassAmherst

Dimensionality Reduction

- In some cases, features might be highly correlated, and therefore redundant
- The number of features can be compressed into a smaller dimensional subspace, reducing storage and increasing performance of the model
- In some cases, dimensionality reduction can lead to better predictive performance if the dataset contains irrelevant features (or noise). In this case, we say that the dataset has a low signal-to-noise ratio

UMassAmherst

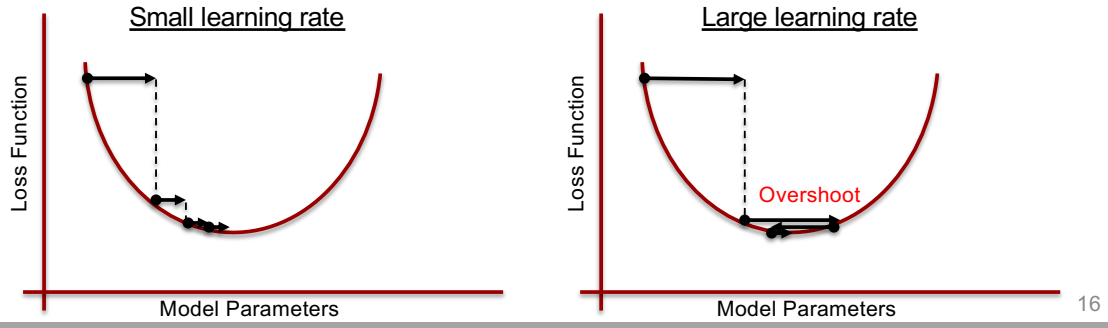
Training and Selecting a Model

- A good practice is to randomly divide our dataset into training and test sets
- A training set is used to train and optimize our model
- A test set is used to evaluate our model

UMassAmherst

Reducing Loss

- Gradient descent
  - the derivative of the loss function with respect to the model parameters.
- Learning rate
  - Multiply the gradient by a scalar



16

16

UMassAmherst

Optimizing computation

- In large data sets, calculating the gradient descent on billions of data points can be very computationally intensive.
- Computing the gradient descent in a small fraction of the data set produces similar results. This is called **Stochastic Gradient** Descent.
- An intermediate solution, is computing the gradient descent in a small batch of data. This approach is called **Mini-Batch Gradient Descent**.

17

17

UMassAmherst

Machine Learning Libraries in Python

- scikit-learn
  - A collection of efficient tools for Machine Learning.
  - http://scikit-learn.org/stable/index.html
- Seaborn
  - Data visualization library built on Matplotlib.
  - https://seaborn.pydata.org/
- TensorFlow
  - An open-source Machine Learning framework developed by Google.
  - https://www.tensorflow.org/
- Pandas
  - A data analysis library
  - https://pandas.pydata.org/
- SciPy
  - An open source library for scientific computing
  - https://www.scipy.org/
- Matplotlib
  - A plotting library
  - https://matplotlib.org/

18

18

UMassAmherst

Managing Your Environment

- Anaconda is a open-source Python distribution for data science and machine learning.
- Anaconda comes with it's own version of a virtual environment (the other being Virtualenv)
- To install Anaconda, head over to https://conda.io/docs/user-guide/install/index.html#regular-installation and follow the instructions according to your operating system
- Test your installation with *$ conda list*

19

19

UMassAmherst

Managing Your Environment

- Create a Python 3 virtual environment using

```
# to create a new environment
conda create -n myenv python=3.6

# to activate and do work
source activate myenv

# to deactivate when done
source deactivate
```

- And manage any packages using
  $ conda install <package name>

More: https://conda.io/docs/_downloads/conda-cheatsheet.pdf

20

20

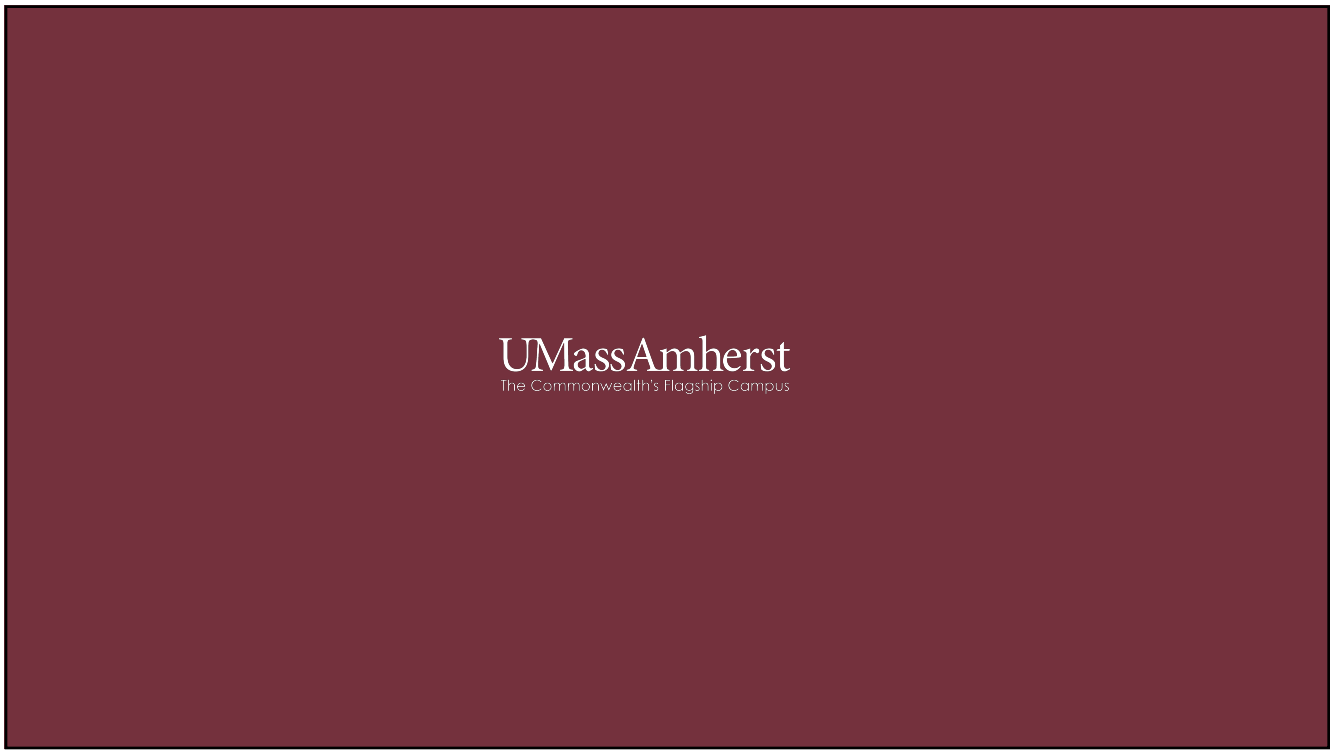
UMassAmherst

Lecture Resources

- https://developers.google.com/machine-learning/crash-course/
- Python Machine Learning, By Sebastian Raschka, Vahid Mirjalili

21

21



22