CS397: Machine Learning Implementation Techniques

What is this course about?

- A broad theoretical and practical understanding of machine learning paradigms and algorithms
- Implementing learning algorithms in C++
- Identifying what algorithm to use based on the type of data
- Understanding the input data

What is this course NOT about?

- Using a tool where the algorithms are implemented
- Algorithms to gather data and how to structure it
- Cleaning and transforming data to apply algorithms on
- Basically we will assume data is already properly formatted for the algorithm to learn

What is Machine Learning?

- Arthur Samuel: “Field of study that gives computers the ability to learn without being explicitly programmed.”
- Construction of algorithms that when given data, can apply categorization, recognize patterns and eventually make predictions on future unseen data
- Tom M. Mitchell’s definition: “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.”

When do we use it?

- Email spam filtering
- Content/Product recommendation
- Face/Voice recognition
- Lunar landing engine control
- Image segmentation
- Text generator
- Self driving cars
- Customer segmentation
- House price deduction
- …

Algorithm Types

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Supervised Learning

- Algorithm is given samples
  - Input
  - Output
- The algorithm will find a relation between the inputs and the outputs
- Example:
  - House price deduction
    - Given data about each house (# of rooms, size, neighborhood, price…)
    - Deduce the output (prices) based on the other values
    - Resulting model will be able to guess the price of a house

Unsupervised Learning

- Algorithm is given samples
  - Inputs
  - No specific output
- Tries to find patterns and gives structure to the data (generates categories)
- Example:
  - Customer segmentation
    - Given data about each customer (products bought, money spent, purchases per week…)
    - Group customers by type (a group might be: frequent buyers that always buy certain products)
    - By inspecting the model we are able to understand the types of customer without prior knowledge

Reinforcement Learning

- No samples are provided
- It is all about states and actions
- States will lead to rewards, the algorithm will try to maximize it by choosing the best action
- Example:
  - Lunar landing (just considering verticality to simplify the problem)
    - States: Height and speed // Actions: Increase speed, reduce speed and do nothing
    - Landing properly will be given a reward, crashing will be penalized
    - Could also add fuel consumption penalty every time speed is adjusted

How do we use those algorithms?

- Supervised & Unsupervised
  - Analyze training data (samples)
  - Build a model
  - Use the model to generate predictions on future data
  - Generalize from examples
  - Incorrect output is clearly identifiable

How do we use those algorithms?

- Reinforcement
  - Agent executes actions and transitions among states
  - Check the feedback (reward)
  - Learn more about the world
  - Generalize from experience
  - Best actions are not known
  - Sub-optimal actions are not explicitly corrected
  - Action is executed and the result was good: Great!
    - Was it the best decision?

Why not explicitly program a solution?

- Sometimes a explicit solution is the way to go
  - ML is not a silver bullet to throw to every problem
  - Explicit solution give forces a real understanding
- Too many features to consider
- Significance of features is unknown
- Too many combinations of actions that could be executed
  - A robot could have 100s of unique actions
  - Any combination of actions could be executed simultaneously
- Features might be inter-dependent
- Example: User signed in to bank account: Valid or Fraud?
  - User used a browser that they haven't used before
  - Username & password entered way quicker or slower than usual
  - Local time at user's location is 3:00am
  - User added a new "transfer to" account

Steps of ML

- Data Collection
- Data Preparation
- Choose a Model (Algorithm)
- Train the Model
- Evaluate the Model
- Parameter Tuning

Data Collection

- The quantity & quality of your data dictate how accurate our model is
- The outcome of this step is generally a representation of data which we will use for training
- Using pre-collected data, by way of datasets from Kaggle, UCI, etc.

Data Preparation

- Prepare data for training
- Clean data: remove duplicates, correct errors, deal with missing values, normalization, data type conversions, etc.
- Randomize data: avoid order due to how data was gathered
- Visualize data: detect relevant relationships between variables or or perform other exploratory analysis
- Split into training and evaluation sets

Choose a Model

- Different algorithms are for different tasks; choose the right one

Train the Model

- The goal of training is to answer a question or make a prediction correctly as often as possible
- Each iteration of process is a training step

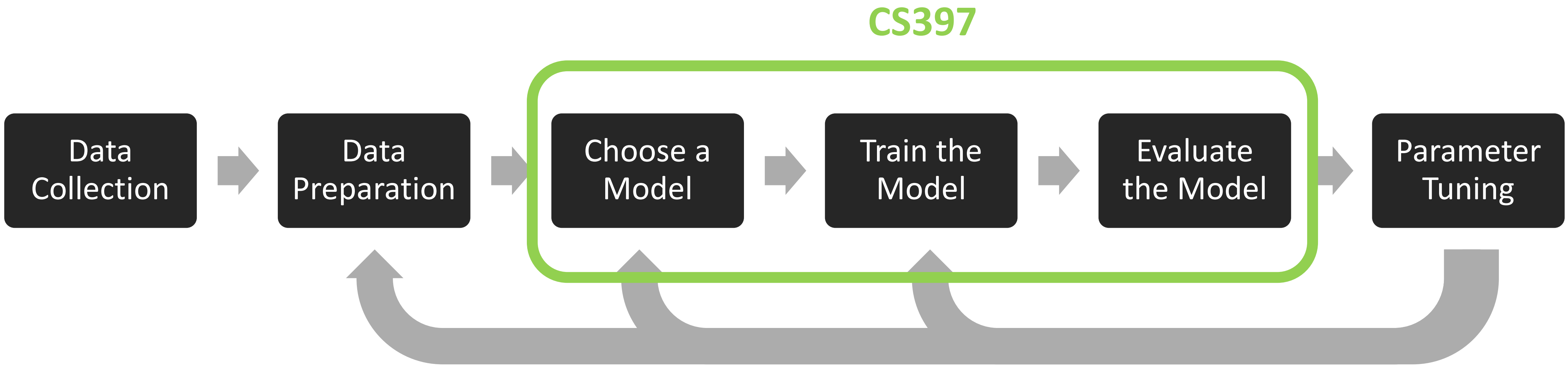
Evaluate the Model

- Uses some metric or combination of metrics to "measure" objective performance of model
- Test the model against previously unseen data
- This unseen data is meant to be somewhat representative of model performance in the real world, but still helps tune the model (as opposed to test data, which does not)

Parameter Tuning

- *Hyperparameter* tuning: readjust the model parameters to improve performance
- Model, regularization, learning rate, number of samples…
- Somehow an art, what you learn through experience and intuition

Steps of ML

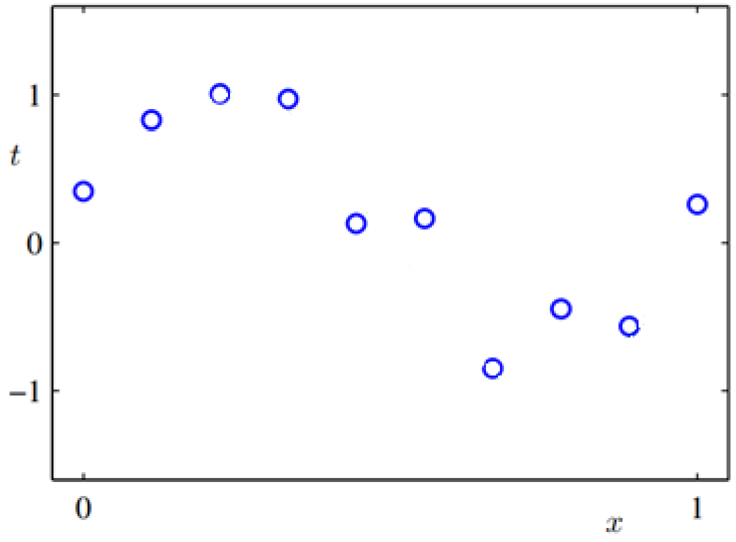

CS397
Data Collection
Data Preparation
Choose a Model
Train the Model
Evaluate the Model
Parameter Tuning

Common Issues: Biased Input

- Personal Bias: Is your data objective?
  - Teachers course evaluations show subjective data
- Special Circumstances Bias: Can the data be generalized?
  - Samples of the impact of a advertisement were collected around a major sports event
- Survivorship bias: Are you missing some type of samples?
  - Evaluating the average DigiPen student by analyzing the graduates
- …

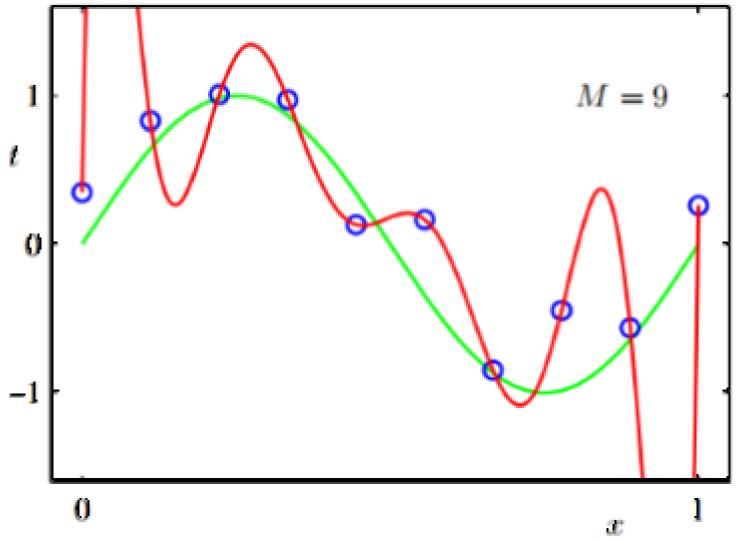
Example: Polynomial Curve Fitting

- Lets create some datapoints from the function $\mathrm{t(x)} = \sin(2\pi\mathrm{x})$ with some added noise
- Green line represents the function
- Blue dots are datapoints (9)



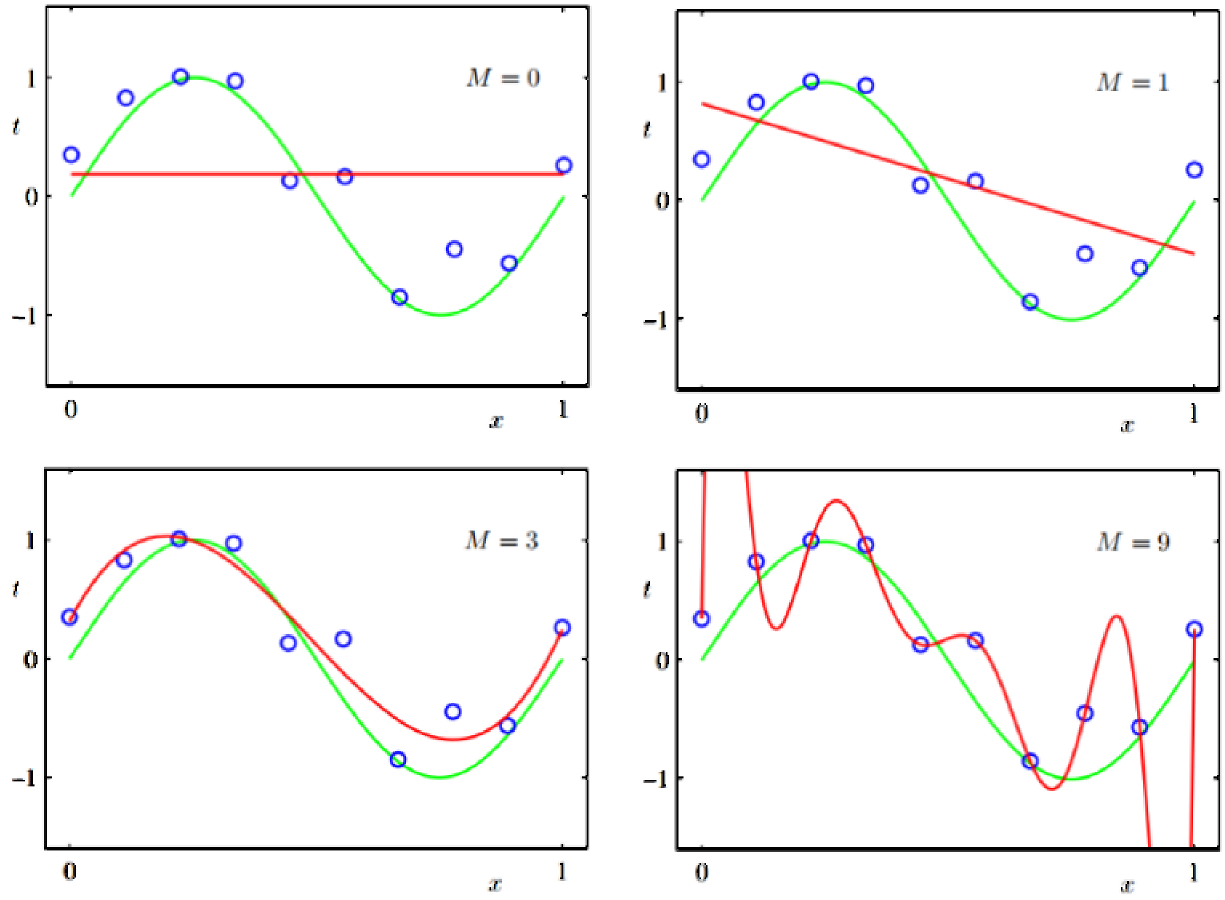
Example: Polynomial Curve Fitting

- We will try to extrapolate the original function so that we can predict values for other values of x. How?
- Lets start with a polynomial.
- What polynomial?
  - Degree 0
  - Degree 1
  - Degree 3
  - Degree 9



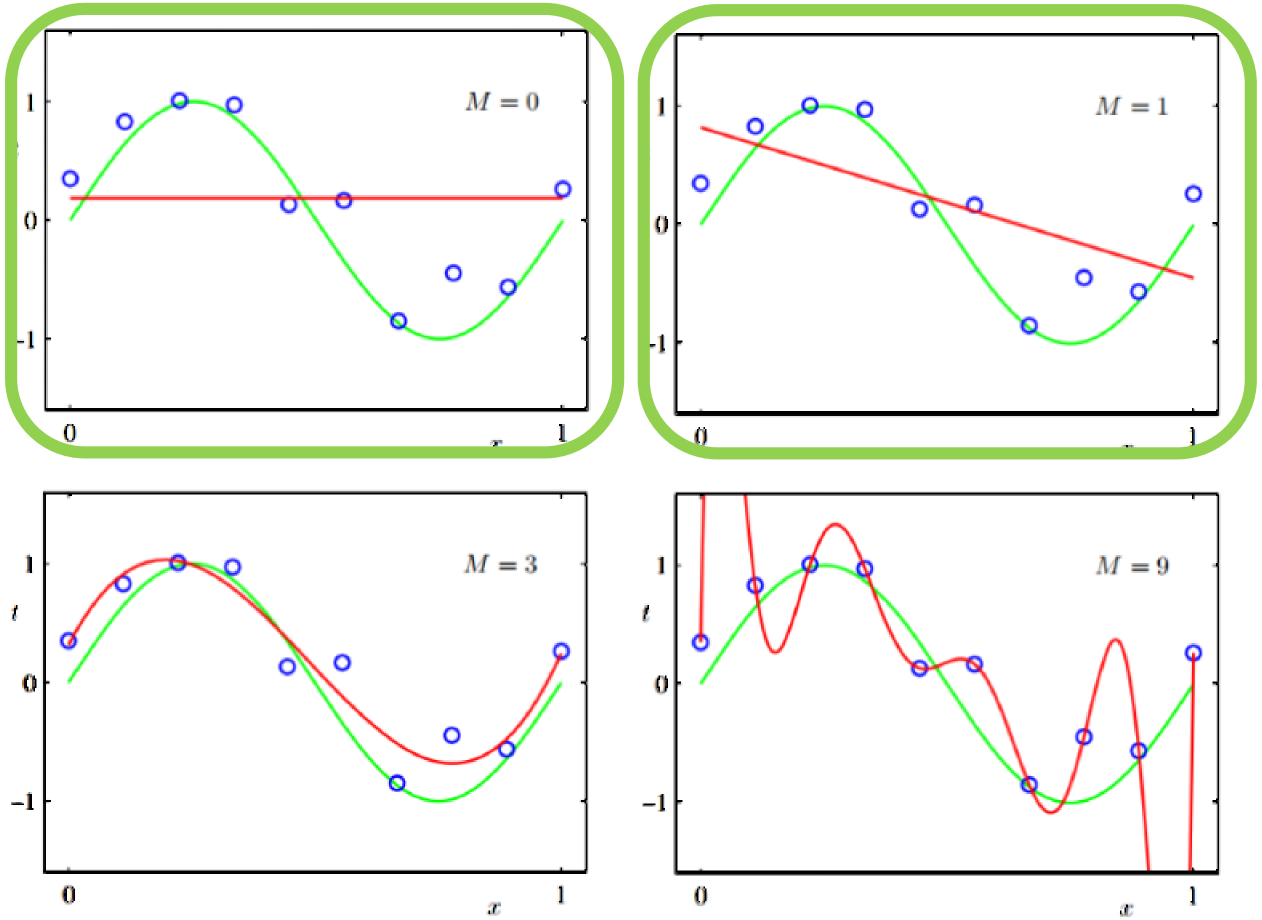
Example: Polynomial Curve Fitting

- How do we compute those red lines?
- Each of those polynomials is generated minimizing the error produced
- This is call **regression**



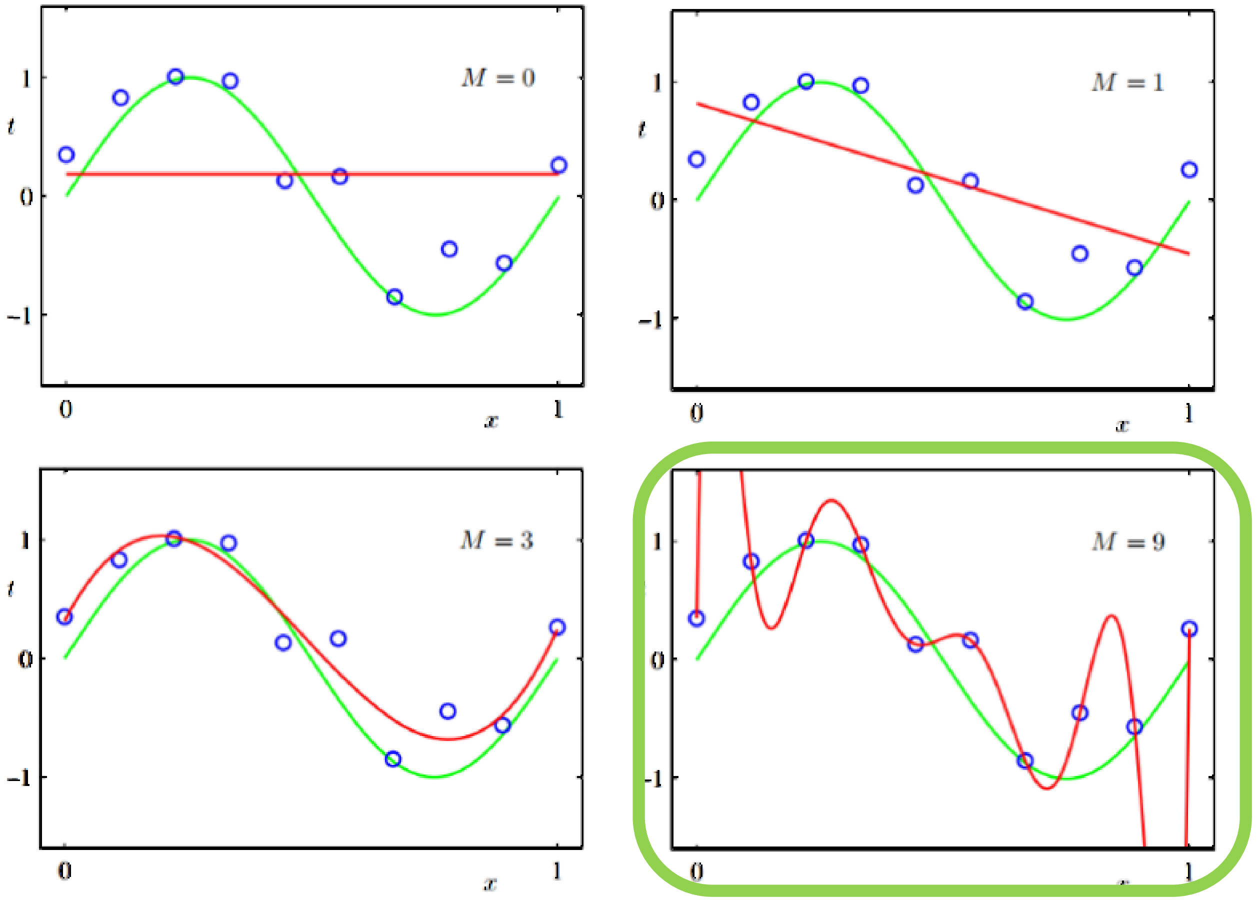
Common Issues: Underfitting

- Model's complexity is less than the complexity of the data
- Model won't even satisfy the training samples
  - Certainly won't work with future data



Common Issues: Overfitting

- Model's complexity is higher than the complexity of the data
- Model will fit the training samples too tightly
  - Won't generalize adequately with future data.



References

- Notes by Antoine Abi Chackra
- https://www.kdnuggets.com/2018/05/general-approaches-machine-learning-process.html
- Pattern Recognition And Machine Learning, by Cristopher M. Bishop