Lecture 1: Overview

Feng Li

Shandong University

fli@sdu.edu.cn

September 6, 2023

Lecture 1: Overview

Instructor

- Prof. Feng Li
- Web: https://funglee.github.io
- Office: N3-312-1
- Education:
  - 2010-2015, PhD, Nanyang Technological University, Singapore.
  - 2007-2010, MS, Shandong University, China.
  - 2003-2007, BS, Shandong Normal University, China.
- Employment:
  - Sep 2022 – Present, Professor, Shandong University, China
  - Sep 2018 – Dec 2021, Associate Professor, Shandong University, China
  - Nov 2015 – Aug 2018, Assistant Professor, Shandong University, China
  - Nov 2014 - Nov 2015, Research Fellow, National University of Singapore, Singapore.
- Research Interests: Distributed Algorithms and Systems, Wireless Networks, Mobile Computing, Internet of Things.

Course Information

- We will investigate fundamental concepts, techniques and algorithms in machine learning. The topics include linear regression, logistic regression, regularization, Gaussian discriminant analysis, Naive Bayes, EM algorithm, SVM, K-means, factor analysis, PCA, neural networks etc.
- 68 hours (4 hours/week $\times$ 17 weeks)
- Labs (35%) + Homework (15%) + Final exam (50%)
- Website: `https://funglee.github.io/ml/ml.html`
- Teaching Assistants (TAs):
    - Ms Lina Wang
    - Mr Fangzheng Duan

Suggested Readings

- Hang Li, Statistical Machine Learning (2nd Ed.), The Tsinghua Press, 2019
- Zhihua Zhou, Machine Learning, Tsinghua Press, 2016
- Tom M. Mitchell, Machine Learning (1st Ed.), China Machine Press, 2008
- Ian Goodfellow, Yoshua Bengio, Deep Learning, People's Posts and Telecommunications Press, 2016
- Trevor Hastie, The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd Ed.), World Publishing Corporation, 2015
- Christopher M. Bishop, Pattern Recognition and Machine Learning (1st Ed.), Springer, 2006

Prerequisite Courses

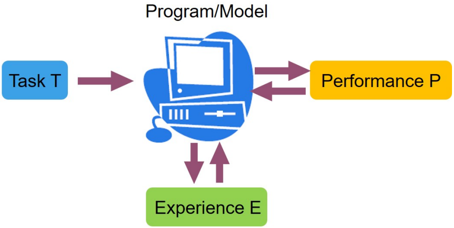
- Linear algebra
- Calculus
- Probability and Statistics
- Information theory
- Convex Optimization

Remarks

- Lectures are important, but not enough.
- You should review what have been taught with more hours than the class hours/weeks.
- You should be familiar with all terminologies related with this course.
- You should understand the mathematical theories behind the machine learning algorithm.
- Practice what you have learned.
- Work hard!

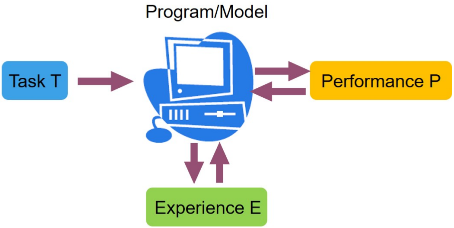
What is Machine Learning ?

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**. [Tom Mitchell, Machine Learning]



What is Machine Learning ? (Contd.)

Improve on task **T**, with respect to performance metric **P**, based on experience **E**.



What is Machine Learning ? (Contd.)

Example 1

- T: Playing checkers
- P: Percentage of games won against an arbitrary opponent
- E: Playing practice games against itself

Example 2

- T: Recognizing hand-written words
- P: Percentage of words correctly classified
- E: Database of human-labeled images of handwritten words

What is Machine Learning ? (Contd.)

Example 3

- T: Categorize email messages as spam or legitimate
- P: Percentage of email messages correctly classified
- E: Database of emails, some with human-given labels

Example 4

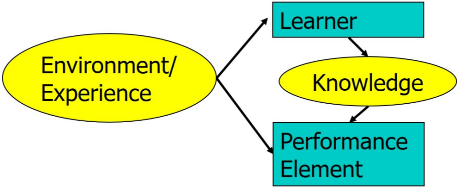
- T: Driving on four-lane highways using vision sensors
- P: Average distance traveled before a human-judged error
- E: A sequence of images and steering commands recorded while observing a human driver

Why Do We Need Machine Learning?

- Develop systems that are too difficult/expensive to construct manually because they require specific detailed skills or knowledge tuned to a specific task (knowledge engineering bottleneck)
- Develop systems that can automatically adapt and customize themselves to individual users.
  - Personalized news or mail filter
  - Personalized tutoring
- Discover new knowledge from large databases (data mining)
  - Market basket analysis (e.g. diapers and beer)
  - Medical information mining (e.g. migraines to calcium channel blockers to magnesium)
- Computational studies of learning may help us understand learning in humans and other biological organisms

Steps to Design a Learning System

- Choose the training experience
- Choose exactly what is to be learned, i.e. the **target function**
- Choose how to represent the target function
- Choose a learning algorithm to infer the target function from the experience.



Training Experience

- **Direct experience**: Given sample input and output pairs for a useful target function.
  - Checker boards labeled with the correct move, e.g. extracted from record of expert play
- **Indirect experience**: Given feedback which is not direct I/O pairs for a useful target function.
  - Potentially arbitrary sequences of game moves and their final results
  - **Credit/Blame Assignment Problem**: How to assign credit blame to individual moves given only indirect feedback?

Source of Training Data

- Provided random examples outside of the learner's control.
  - Negative examples available or only positive?
- Good training examples selected by a "benevolent" teacher.
  - "Near miss" examples
- Learner can query an oracle about class of an unlabeled example in the environment
- Learner can construct an arbitrary example and query an oracle for its label
- Learner can design and run experiments directly in the environment without any human guidance.

Applications of Machine Learning

Document Search

- Given counts of words in a document, determine what its topic is.
- Group documents by topic without a pre-specified list of topics.
- Many words in a document, many, many documents available on the web.

Image/Video Understanding

- Given an/a image/video, determine what objects it contains.
- Determine what semantics it contains
- Determine what actions it contains.

Applications of Machine Learning (Contd.)

Cancer Diagnosis

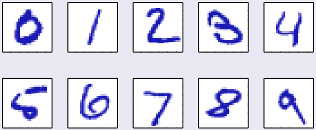
- Given data on expression levels of genes, classify the type of tumor.
- Discover categories of tumors having different characteristics.

Marketing

- Given data on age, income, etc., predict how much each customers spends.
- Discover how the spending behaviors of customers are related.
- Fair amount of data on each customer, but messy
- May have data on a very large number of customer.

Example 1: Handwritten Digit Recognition

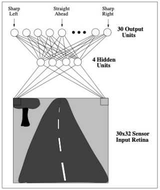
- Handcrafted rules will result in large number of rules and exceptions
- Better to have a machine that learns from a large training set
- Handwriting recognition cannot be done without machine learning!

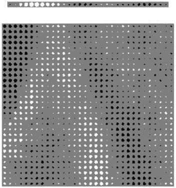


Everyone has different writing style!

Example 2: Autonomous Driving-ALVINN

- A predecessor of Google car drives 70 mph on a public highway

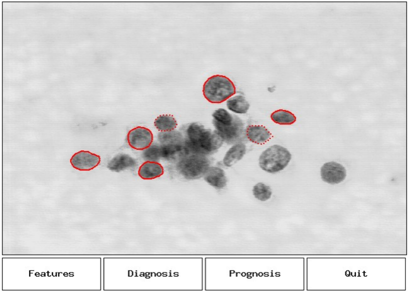


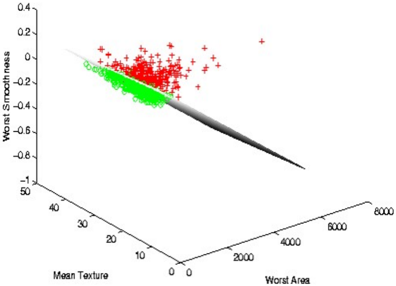


30x32 weights into one out of four hidden unit

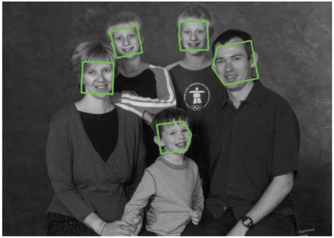
- 30 outputs for steering
- 4 hidden units
- 30x32 pixels as inputs

Example 3: Breast Cancer Diagnosis





Example 4: Face Recognition



Categories of Machine Learning

1. Supervised learning: learning with a teacher
   - Training examples with labels are given
2. Unsupervised learning: learning without a teacher
   - Training examples without labels.
3. Reinforcement Learning: learning by interacting
4. Semi-supervised learning: partially supervised learning
5. Active learning: actively making queries

Supervised Learning

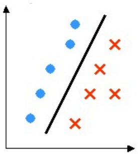
- In the ML literature, a supervised learning problem has the following characteristics:
  - We are primarily interested in prediction
  - We are interested in predicting only one thing
  - The possible values of what we want to predict are specified, and we have some training cases for which its value is known
- The thing we want to predict is called the **target** or the **response variable**
- Usually, we need training data

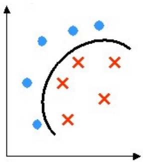
Supervised Learning

- For **classification** problem, we want to predict the class of an item.
  - The type of tumor, the topic of a document, the semantics contained in an image, whether a customer will purchase a product.
- For a **regression** problem, we want to predict a numerical quantity.
  - The amount of customer spends, the blood pressure of a patient, etc.
- To make predictions, we have various inputs,
  - Gene expression levels for predicting tumor type, age and income for predicting amount spent, the features of images with known semantics

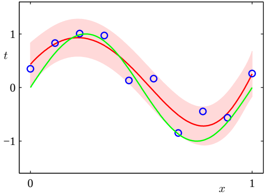
Classification and Regression

- Classification: finding decision boundaries





- Regression: fitting a curve/plane to data



Supervised Classification Problems

- Cancer diagnosis (Training Set)

Patient ID	# of Tumors	Avg Area	Avg Density	Diagnosis
1	5	20	118	Malignant
2	3	15	130	Benign
3	7	10	52	Benign
4	2	30	100	Malignant

- Use the above training set to learn how to classify patients where diagnosis is not known (Test Set):

Patient ID	# of Tumors	Avg Area	Avg Density	Diagnosis
101	4	16	95	?
102	9	22	125	?
103	1	14	80	?

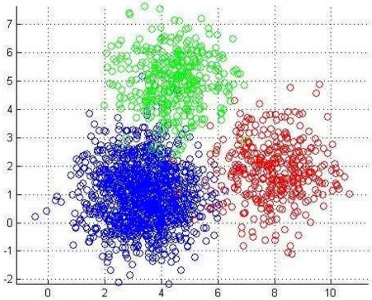
Supervised Regression Problems

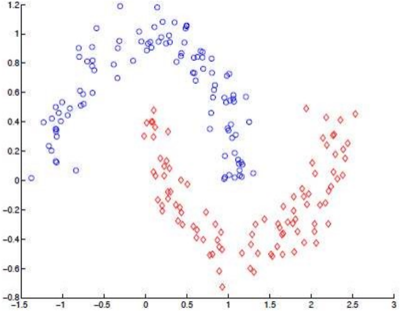
- Predict tomorrow's stock market price given current market conditions and other possible side information
- Predict the age of a viewer watching a given video on YouTube
- Predict the location in 3D space of a robot arm end effector, given control signals (torques) sent to its various motors
- Predict the amount of prostate specific antigen (PSA) in the body as a function of a number of different clinical measurements.
- Predict the temperature at any location inside a building using weather data, time, door sensors, etc.

Unsupervised Learning

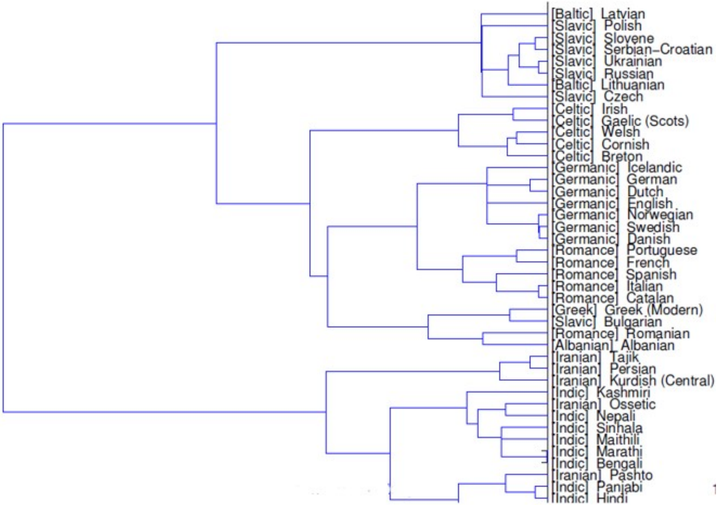
- For an unsupervised learning problem, we do not focus on prediction of any particular thing, but try to find interesting aspects of the data
- Examples:
  - Discovering clusters
  - Discovering latent factor
  - Discovering graph structure
  - Matrix completion

Unsupervised Learning: Discovering Clusters





Unsupervised Learning: Clustering (Contd.)



Unsupervised Learning: Discovering Latent Factors

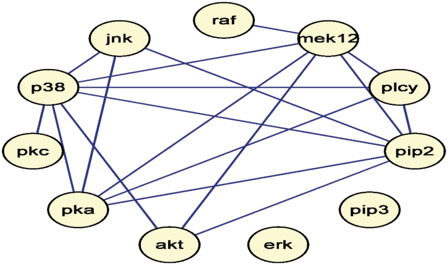
- **Dimensionality reduction**
  - When dealing with high dimensional data, it is often useful to reduce the dimensionality by projecting the data to a lower dimensional subspace which captures the “essence” of the data.
  - The motivation behind this technique is that although the data may appear high dimensional, there may only be a small number of degrees of variability, corresponding to **latent factors**

(a)

(b)

Unsupervised Learning: Discovering Graph Structures

- Sometimes we measure a set of correlated variables, and we would like to discover which ones are most correlated with which others
- This can be represented by a graph, in which nodes represent variables, and edges represent direct dependence between variables

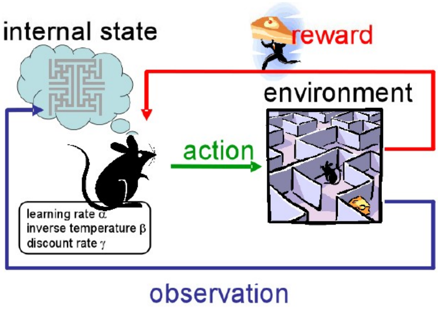


Unsupervised Learning: Matrix Completion

- Sometimes we have missing data, that is, variables whose values are unknown, such that the corresponding design matrix will then have “holes” in it
- The goal of **matrix completion** is to infer plausible values for the missing entries

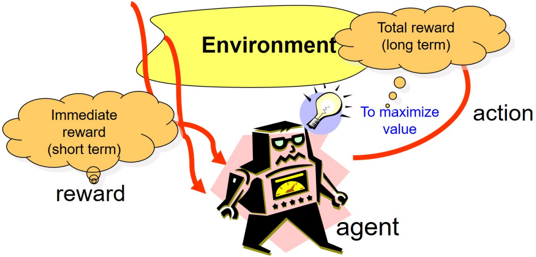
Reinforcement Learning

- Learning from interaction (with environment)
- Goal-directed learning
- Learning **what to do** and its **effect**
- **Trial-and-error** search and **delayed reward**



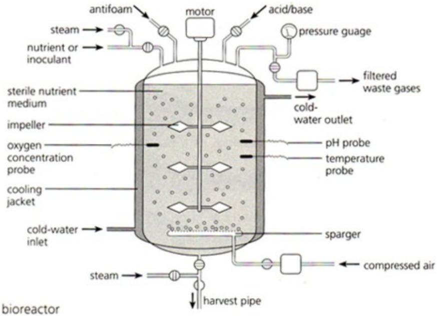
Reinforcement Learning (Contd.)

- The agent has to exploit what it already knows in order to obtain reward, but it also has to explore in order to make better action selections in the future.
- Dilemma: neither exploitation nor exploration can be pursued exclusively without failing at the task.



Reinforcement Learning (Contd.)

- Example (Bioreactor)
- State
  - Current temperature and other sensory readings, composition, target chemical
- Actions
  - How much heating, stirring are required?
  - What ingredients need to be added?
- Reward
  - Moment-by-moment production of desired chemical



A bioreactor

Reinforcement Learning (Contd.)

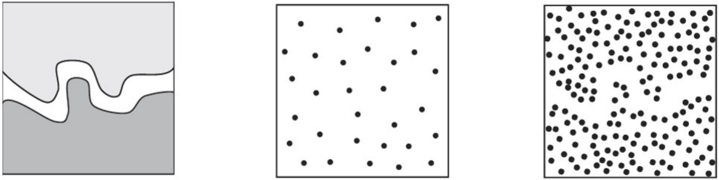
- Example (Pick-and-Place Robot)
- State
  - Current positions and velocities of joints
- Actions
  - Voltages to apply to motors
- Reward
  - Reach end-position successfully, speed, smoothness of trajectory

Reinforcement Learning (Contd.)

- Example (Recycling Robot)
- State
  - charge level of battery
- Actions
  - Look for cans, wait for can, go recharge
- Reward
  - Positive for finding cans, negative for running out of battery

Semi-supervised Learning

- As the name suggests, it is in between Supervised and Unsupervised learning techniques w.r.t the amount of labeled and unlabeled data required for training.
- With the goal of reducing the amount of supervision required compared to supervised learning.
- At the same time, improving the results of unsupervised clustering to the expectations of the user.



With lots of unlabeled data the decision boundary becomes apparent.

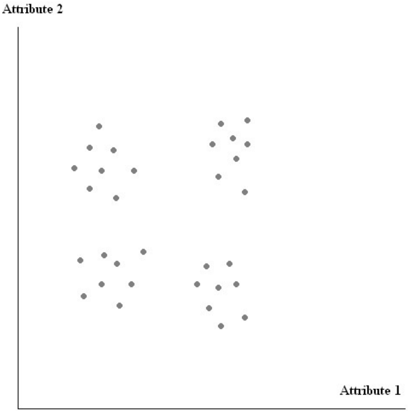
Semi-supervised Learning (Contd.)

- Constrained Clustering
- Distance Metric Learning
- Manifold based Learning
- Sparsity based Learning (Compressed Sensing)

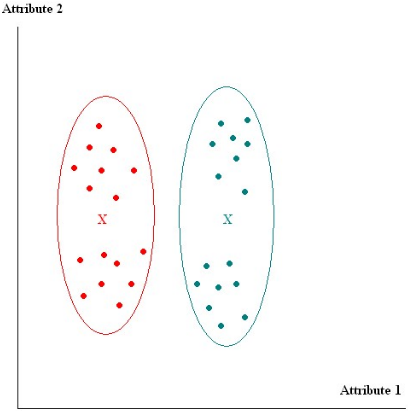
Constrained Clustering

- When we have any of the following:
  - Class labels for a subset of the data.
  - Domain knowledge about the clusters.
  - Information about the 'similarity' between objects.
  - User preferences.
- May be pairwise constraints or a labeled subset.
  - Must-link or cannot-link constraints.
  - Labels can always be converted to pairwise relations.
- Can be clustered by searching for partitioning that respect the constraints
- Recently the trend is toward similarity-based approaches

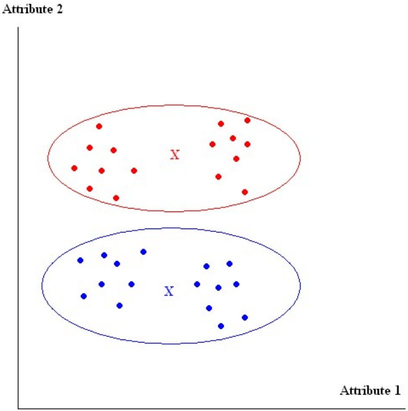
Constrained Clustering (Contd.)



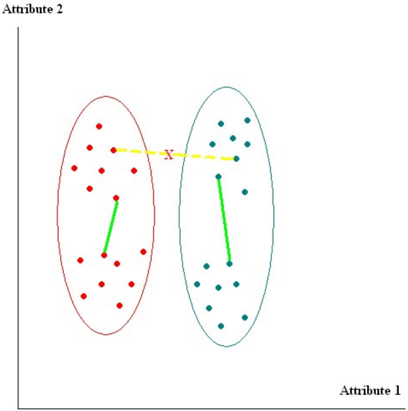
Constrained Clustering (Contd.)



Constrained Clustering (Contd.)



Constrained Clustering (Contd.)

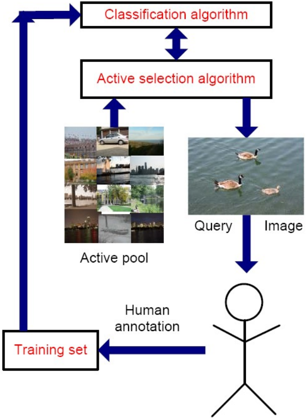


Active Learning

- Basic idea:
  - Traditional supervised learning algorithms passively accept training data.
  - Instead, query for annotations on informative images from the unlabeled data.
  - Theoretical results show that large reductions in training sizes can be obtained with active learning!
- But how to find images that are the most informative?

Active Learning (Contd.)

- One idea uses uncertainty sampling.
- Images on which you are uncertain about classification might be informative!
- What is the notion of uncertainty?
  - Idea: Train a classifier like SVM on the training set.
  - For each unlabeled image, output probabilities indicating class membership.
  - Estimate probabilities can be used to infer uncertainty.
  - A one-vs-one SVM approach can be used to tackle multiple classes.



Parametric vs Non-Parametric Models

- Parametric model
  - We can train a model by using the training data to estimate parameters of it
  - Use these parameters to make predictions for the test data.
  - Such approaches save computation when we make predictions for test data.
  - That is, estimate parameters once, use them many times,
  - E.g. Linear regression

$$\hat{y} = \hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j x_j$$

Parametric vs Non-Parametric Models (Contd.)

- Non-parametric model: nearest-neighbor method
  - Make predictions for test data based on a subset of training cases, e.g., by approximating the mean, median or mode of $p(y|x)$.

$$\hat{y} = \frac{1}{K} \sum_{i \in N_K(x)} y_i$$

  - Big question: How to choose K?
  - If K is too small, we may "overfitting", but if K is too big, we will average over training cases that aren't relevant to the test case.

Parametric vs Non-Parametric Models (Contd.)

- These two methods are opposite w.r.t. computation.
  - NN-like methods are memory-based methods. We need to remember all the training data.
  - Linear regression, after getting parameters, can forget the training data, and just use the parameters.
- They are also opposite w.r.t. to statistical properties.
  - NN makes few assumptions about the data, and has a high potential for over-fitting.
  - Linear regression makes strong assumption about the data, and consequently has a high potential for bias.

The Curse of Dimensionality

- Handling complexity
  - Involve many variables, how can we handle this complexity without getting into trouble.
- Optimization and Integration
  - Usually involve finding the best values for some parameters (an optimization problem), or average over many plausible values (an integration problem). How can we do this efficiently when there are many parameters.
- Visualization
  - Understanding what's happening is hard, 2D? 3D?
- All these challenges become greater when there are many variables or parameters —-the so-called "curse of dimensionality".
  - But more variables also provide more information
  - A blessing? A curse?

How to Handle Complexity

- Properly dealing with complexity is a crucial issue for machine learning.
- Limiting complexity is one approach
  - Use a model that is complex enough to represent the essential aspects of the problem, but that is not so complex that overfitting occurs.
  - Overfitting happens when we choose parameters of a model that fit the data we have very well, but do poorly on new data (poor generalization ability).
  - Cross-validation, regularization,
- Reducing dimensionality is another possibility.
  - It is apparent that things become simpler if can find out how to reduce the large number of variables to a small number.
- Averaging over complexity is the Bayesian approach.
  - Use as complex a model might be needed, but don't choose a single parameter values. Instead, average the predictions found using all the parameter values that fit the data reasonably well, and which are plausible for the problem

Example of Complexity

Plots of polynomials having various degree, shown as red curves.

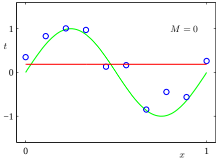


Figure: Degree = 0

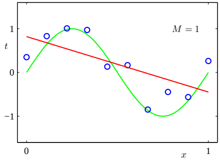


Figure: Degree = 1

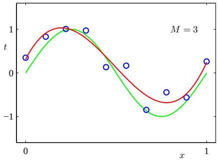


Figure: Degree = 3

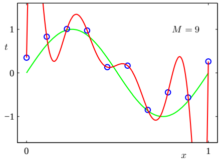
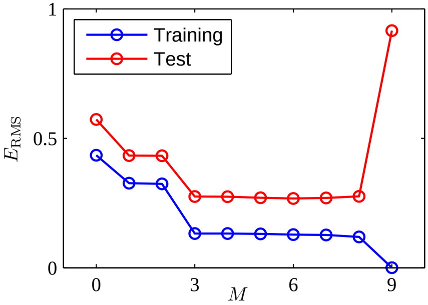


Figure: Degree = 9

Example of Complexity

Graphs of the root-square error, evaluated on the training set and on an independent test set for various degree.



Does Complexity should be limited?

- If we make predictions using "the best" parameters of a model, we have to limit the number of parameters to avoid over-fitting.
- For this example, the model with degree=3 seems good. We might be able to choose a good value for M using the method of "cross validation", which looks for the value that does best at prediction one part of the data from the rest of the data.
- But we know $\sin(2\pi x)$ is not a polynomial function, it has an infinite series representation with terms of arbitrarily high order.
- How can it be good to use a model that we know is false?
  - The Bayesian answer: It is not good. We should abandon the idea of using the best parameters and instead average over all plausible values for the parameters. Then we can use a model (perhaps a very complex one) that is as close to being correct as we can manage.

Reducing Dimensionality

- Suppose dimension of input data is 1000, can we replace these with fewer ones, without loss of information.
- On simple way is to use PCA (Principal Component Analysis)
    - Suppose that all data are in a space, we first find the direction of highest variance of these data points, then the direction of second-highest variance that is orthogonal to the first one, so on and so forth¡
    - Replace each training sample by the projections of the inputs on some directions.
- Might discard useful info., but keep most of it.

Thanks!

Q & A