

# CSCE 50603-001

# Machine Learning

Fall 2025


# Overview

- Class hour MoWeFr 10:45 - 11:35AM.
  - Location: JBHT 239
- Office hour MoWe 2:00 - 3:00PM or by appointment.
  - Location: JBHT 522
- Instructor – Lu Zhang
  - Email: [lz006@uark.edu](mailto:lz006@uark.edu)
  - Office: JBHT 522
  - Webpage: <https://zh0007lu.github.io/>
- Course Website
  - <https://zh0007lu.github.io/course/2025fall/50603/50603.html>

# Course Material

- No required textbook.
- Reference materials:
  - The Elements of Statistical Learning, by Trevor Hastie, et. al. (2009)
    - Available online: <https://web.stanford.edu/~hastie/ElemStatLearn/>
  - Machine Learning: a Probabilistic Perspective, by Kevin Murphy (2012)
  - Understanding Machine Learning: From Theory to Algorithms, by Shai Shalev-Shwartz and Shai Ben-David (2014)
    - Available online: <https://www.cse.huji.ac.il/~shais/UnderstandingMachineLearning/>
  - Dive into Deep Learning, by Aston Zhang and Zachary C. Lipton and Mu Li and Alexander J. Smola (2020)
    - Available online: <https://d2l.ai/>

# Course Prerequisite

- CSCE graduate standing
- Expect that students should know/have
  - Linear algebra
  - Calculus
  - Probability and statistics

Basic concepts
- Good programming skills in at least one of Python, Java or Matlab
  - Python is highly recommended

# Grading

- Composition
  - Assignment 30%
  - Midterm 15%
  - Group project 30%
  - Final 25%
- The final class grade will be assigned according to the 10-point scale shown below. The grades may or may not be curved.
  - A 90 – 100%
  - B 80 – 89.9%
  - C 70 – 79.9%
  - D 60 – 69.9%
  - F < 60%

# Assignment

- There will be 3 assignments that will enhance understanding of material taught in the course.
- The assignment requirements and due dates will be posted on the course website.
- Student should NOT use any ML libraries.
- Assignments must be submitted electronically through Blackboard by 11:59 pm of the due date specified in the assignment description.
- Late policy
  - 10% penalty for each day after the due date for up to 3 days late.
  - Assignment more than 3 days late should be submitted together with an explanation.
  - Weekends count as 1 day.

# Group Project

- There will be one group project that will deepen your exploration of machine learning with real-world data.
  - 1-3 students per group.
- The project requirements, possible topics and due date will be posted on the course website.
- Students CAN use any ML libraries or materials from the Internet.
- Project presentation before end of semester.
- A project report is required.

# Exams

- Two exams: midterm and final.
- For both exams, students ARE allowed one 8.5x11 page of hand-written or printed notes (double-sided) and a calculator, but they are NOT allowed any other materials or other electric devices such as cell phones, smart watches, tablets, or computers.
- Both exams will be conducted physically.



# Office Hours

- Office hours will be primarily conducted physically at JBHT 522
- Students can also request virtual meetings using Zoom or Blackboard Ultra.

# University Policies

- Academic Integrity
  - Refer to <https://honesty.uark.edu/policy/>
- Emergency Preparedness
  - Refer to <http://emergency.uark.edu/>
- Inclement Weather
  - Refer to <http://safety.uark.edu/inclement-weather/>
- RazALERT
  - Refer to <http://safety.uark.edu/emergency-preparedness/emergency-notification-system/>
- Academic Support
  - Refer to <http://www.uark.edu/academics/academic-support.php>

# Academic Dishonesty Policy

- As a core part of its mission, the University of Arkansas provides students with the opportunity to further their educational goals through programs of study and research in an environment that promotes freedom of inquiry and academic responsibility. Accomplishing this mission is only possible when intellectual honesty and individual integrity prevail. Each University of Arkansas student is required to be familiar with and abide by the University's 'Academic Integrity Policy' at [honesty.uark.edu](https://honesty.uark.edu). Students with questions about how these policies apply to a particular course or assignment should immediately contact their instructor.

# About Generative AI

- There is no university policy forbidding the use of AI generative tools
- Advice for this class
  - Limited use for the assignments
  - Free to use for the project
  - Document the use of AI

# Introduction to Machine Learning

Adopted from slides by Geoffrey Hinton, Andrew Ng, and Pedro Domingos

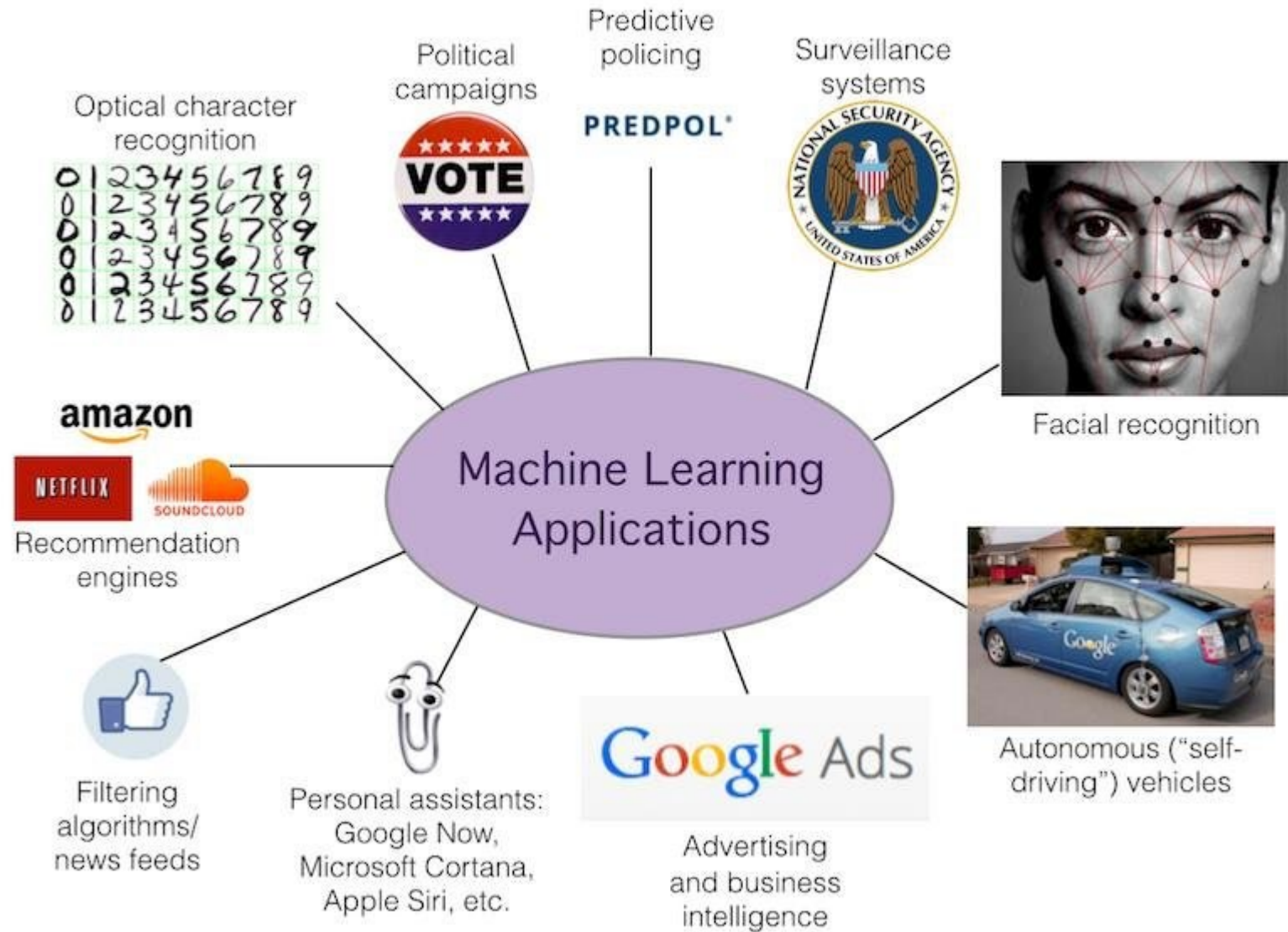
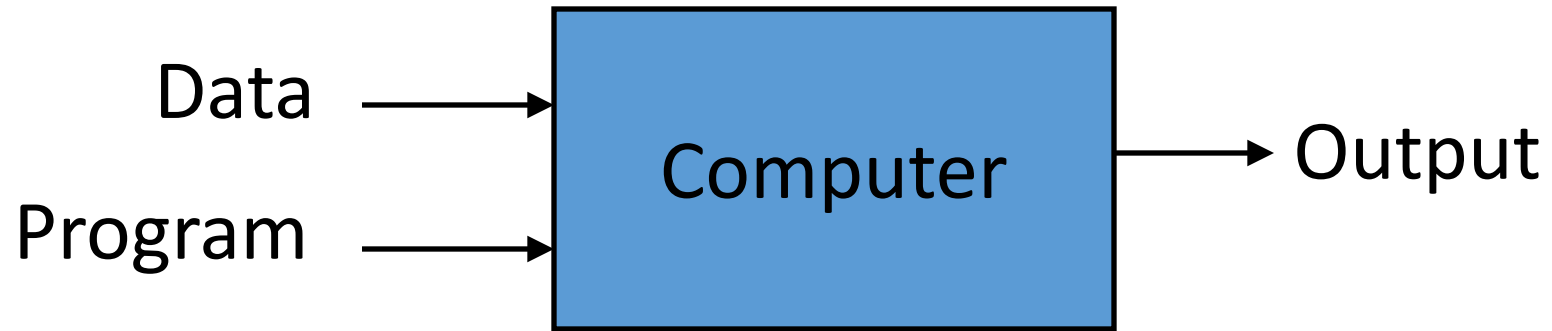


Figure from Ahmad F. Al Musawi

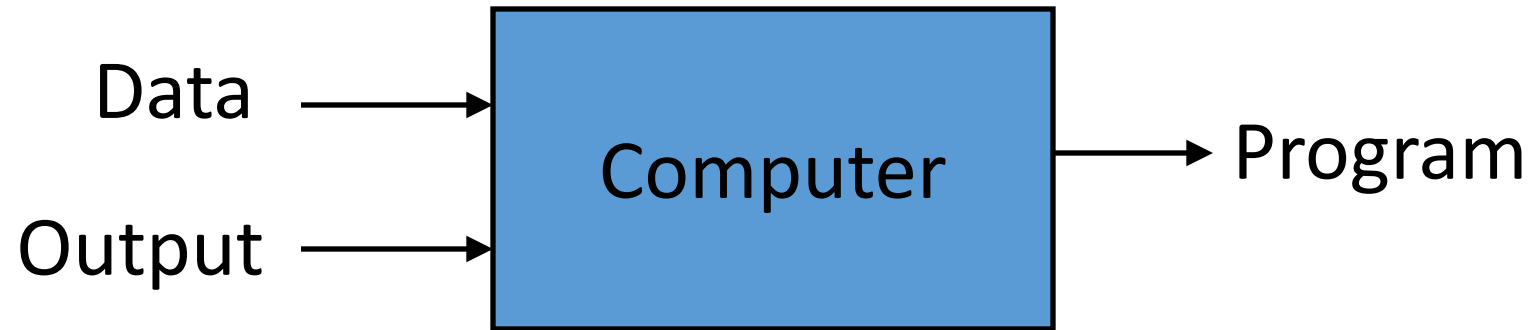
# What Is Machine Learning?

- It is very hard to write programs that solve problems like recognizing a face.
  - We don't know what program to write because we don't know how our brain does it.
  - Even if we had a good idea about how to do it, the program might be horrendously complicated.
- Instead of writing a program by hand, we collect lots of examples that specify the correct output for a given input.
- A machine learning algorithm then takes these examples and produces a program that does the job.
  - The program produced by the learning algorithm may look very different from a typical hand-written program. It may contain millions of numbers.
  - If we do it right, the program works for new cases as well as the ones we trained it on.

## Traditional Programming



## Machine Learning





# Types of Learning Task

- **Supervised learning**
  - Training data includes desired outputs
- **Unsupervised learning**
  - Training data does not include desired outputs
- **Semi-supervised learning**
  - Training data includes a few desired outputs
- **Self-supervised learning**
  - Training data contains supervised signals extracted from the data itself
- **Reinforcement learning**
  - Rewards from sequence of actions

# Supervised Learning

- **Given** examples of a function  $(X, Y=F(X))$
- **Estimate** function  $F(X)$  to predict  $Y$  for new examples  $X$ 
  - Discrete  $Y$ : Classification
  - Continuous  $Y$ : Regression
  - $F(X) = \text{Probability}(X)$ : Probability estimation

# Unsupervised Learning

- **Given** examples ( $X$ )
- **Understand** patterns of  $X$ 
  - How each example is related to one another

# What We'll Cover

- **Supervised learning**

- Linear regression
- Decision tree
- Naïve bayes
- Instance-based learning
- Logistic regression
- Support vector machines
- Neural networks
- Gradient descent-based optimization
- PAC Learning theory

- **Unsupervised learning**

- Clustering
- Latent variable model

- **Advanced topic**

- Fairness-aware machine learning
- Deep learning
- Reinforcement learning
- Causal modeling and inference

# ML in a Nutshell

- Tens of thousands of machine learning algorithms
- Hundreds new every year
- Every machine learning algorithm has three components:
  - **Representation**
  - **Evaluation**
  - **Optimization**

# Representation

- Decision trees
- Sets of rules / Logic programs
- Instances
- Graphical models (Bayes/Markov nets)
- Neural networks
- Support vector machines
- Ensemble models
- Etc.

# Evaluation

- Accuracy
- Precision and recall
- Squared error
- Likelihood
- Posterior probability
- Cost / Utility
- Margin
- Entropy
- K-L divergence
- Etc.

# Optimization

- Combinatorial optimization
  - E.g.: Greedy search
- Convex optimization
  - E.g.: Gradient descent
- Constrained optimization
  - E.g.: Linear programming



# Supervised Learning

- **Given** examples of a function  $(X, Y=F(X))$
- **Estimate** function  $F(X)$  to predict  $Y$  for new examples  $X$ 
  - Discrete  $Y$ : Classification
  - Continuous  $Y$ : Regression
  - $F(X) = \text{Probability}(X)$ : Probability estimation

# Representation - Hypothesis Space

- One way to think about a supervised learning machine is as a device that explores a “hypothesis space”.
  - Each setting of the parameters in the machine is a different hypothesis about the function that maps input vectors to output vectors.
- The art of supervised machine learning is in:
  - Deciding how to represent the inputs and outputs
  - Selecting a hypothesis space that is powerful enough to represent the relationship between inputs and outputs but simple enough to be searched.

## Supervised Learning

**Given** examples of a function  $(X, Y = F(X))$

**Find** an estimation of function  $F(X)$  from hypothesis space  $\mathcal{H}$

# Evaluation - Loss Functions

- Mean Square Error (MSE): Squared difference between actual and target real-valued outputs.

$$MSE = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}$$

- Cross Entropy/Negative Log Likelihood: Multiplying the log of the actual predicted probability for the ground truth class

$$CrossEntropy = -(y_i \log(p_i) + (1 - y_i) \log(1 - p_i))$$

- Hinge Loss
- K-L Divergence

# Optimization - Searching a hypothesis space

- The obvious method is to first formulate a loss function and then adjust the parameters to minimize the loss function.
  - Gradient descent
- Bayesians do not search for a single set of parameter values that do well on the loss function.
  - They start with a prior distribution over parameter values and use the training data to compute a posterior distribution over the whole hypothesis space.
  - Markov Chain Monte Carlo (MCMC)

# Generalization

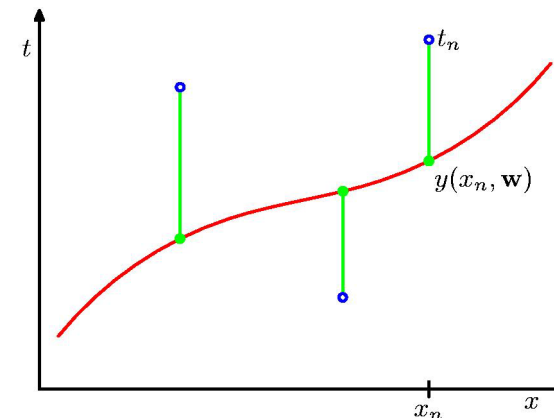
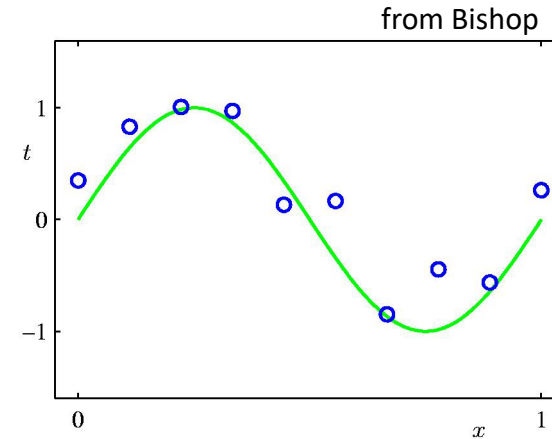
- The real aim of supervised learning is to do well on test data that is not known during learning.
- Choosing the values for the parameters that minimize the loss function on the training data is not necessarily the best policy.
- We want the learning machine to model the true regularities in the data and to ignore the noise in the data.
  - But the learning machine does not know which regularities are real and which are accidental quirks of the particular set of training examples we happen to pick.
- So how can we be sure that the machine will generalize correctly to new data?

# Trading off the goodness of fit against the complexity of the model

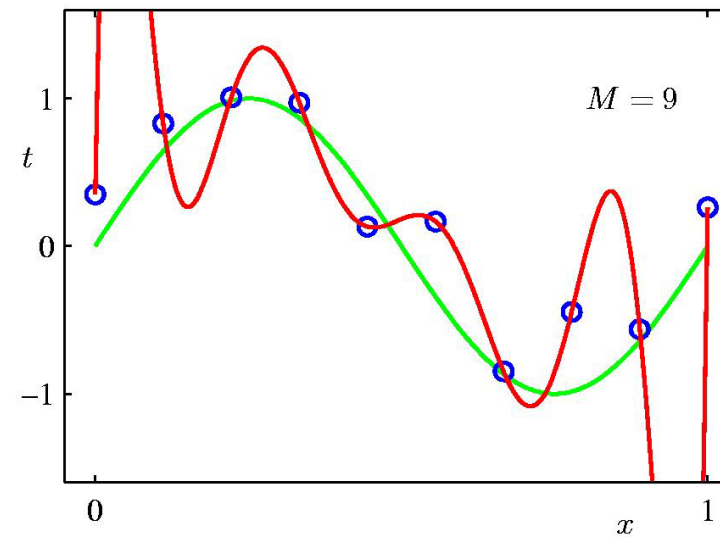
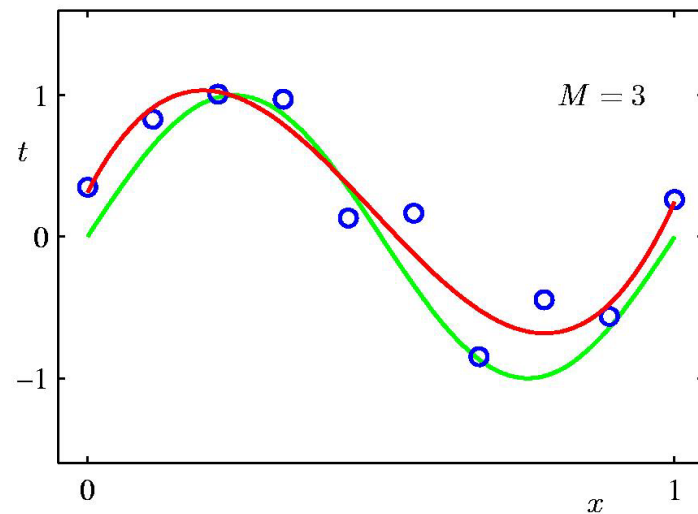
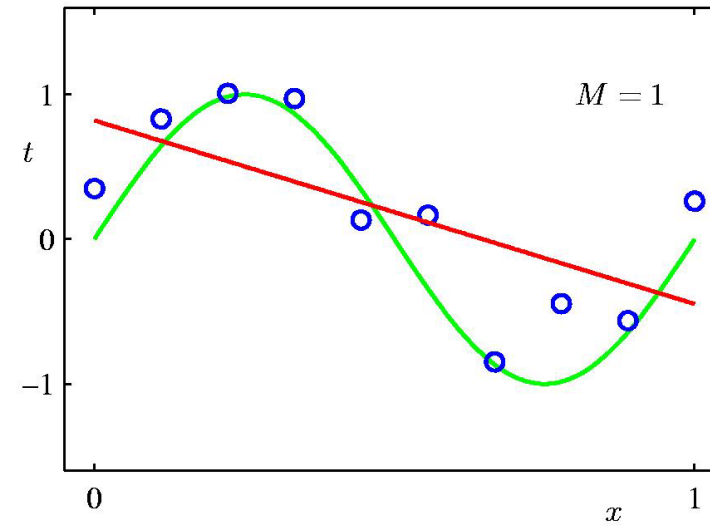
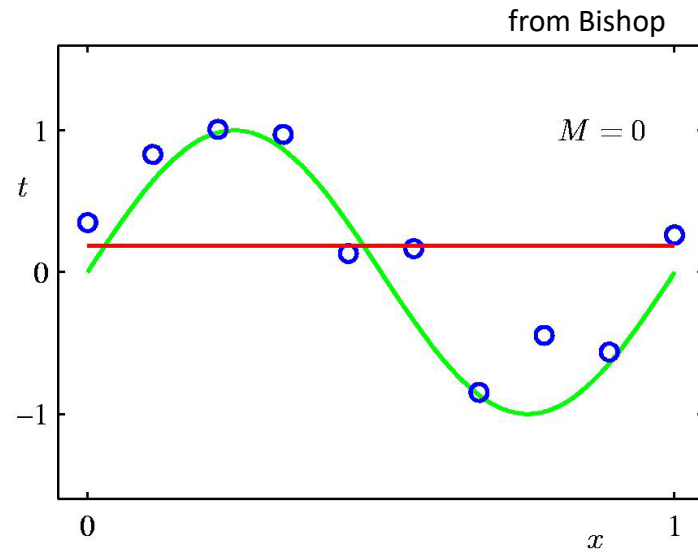
- It is intuitively obvious that you can only expect a model to generalize well if it explains the data surprisingly well given the complexity of the model.
- If the model has as many degrees of freedom as the data, it can fit the data perfectly but so what?
- There is a lot of theory about how to measure the model complexity and how to control it to optimize generalization.
  - Some of this “learning theory” will be covered later in the course, but it requires a whole course on learning theory to cover it properly

# A simple example: Fitting a polynomial

- The green curve is the true function (which is not a polynomial)
- The data points are uniform in  $x$  but have noise in  $y$ .
- We will use a loss function that measures the squared error in the prediction of  $y(x)$  from  $x$ . The loss for the red polynomial is the sum of the squared vertical errors.

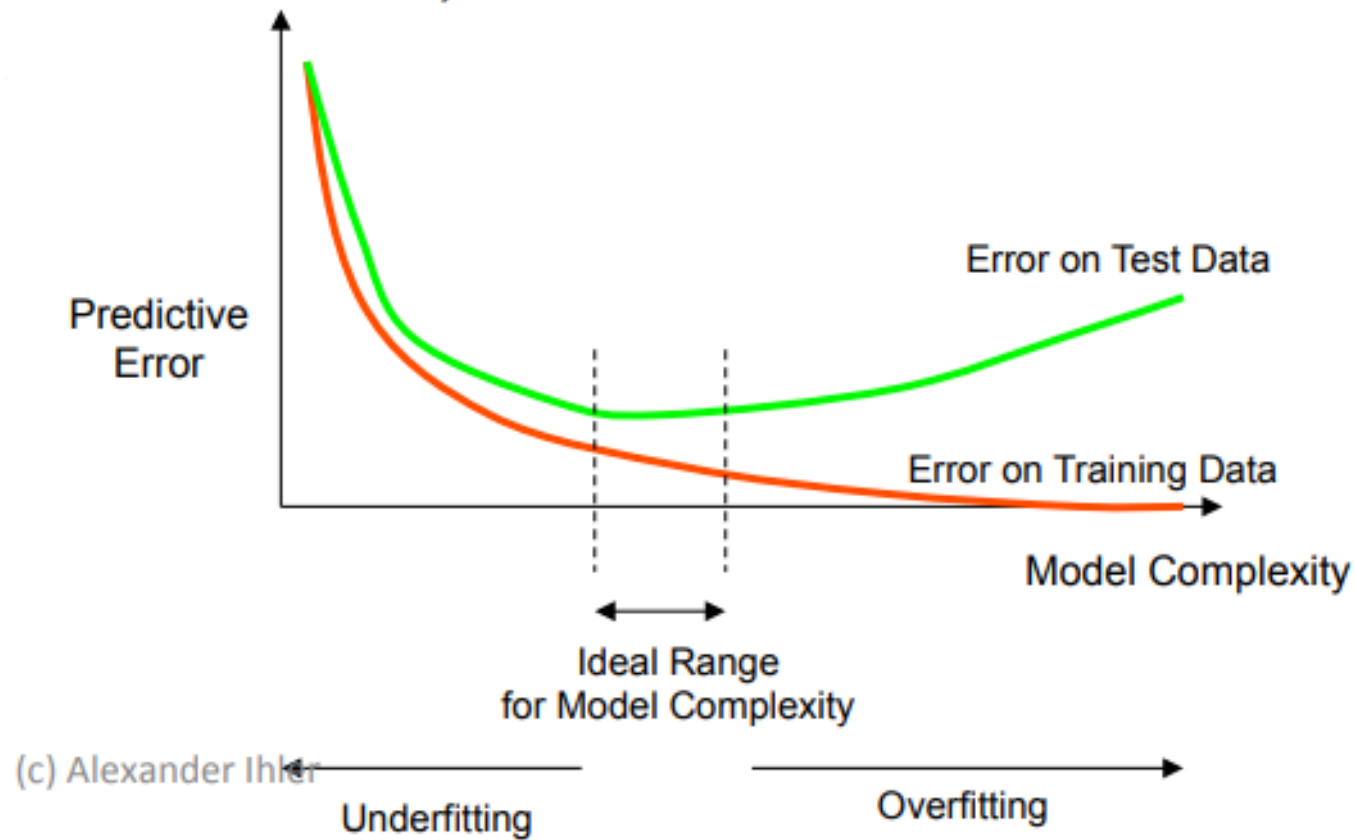


# Some fits to the data: which is best?





# Underfitting and overfitting



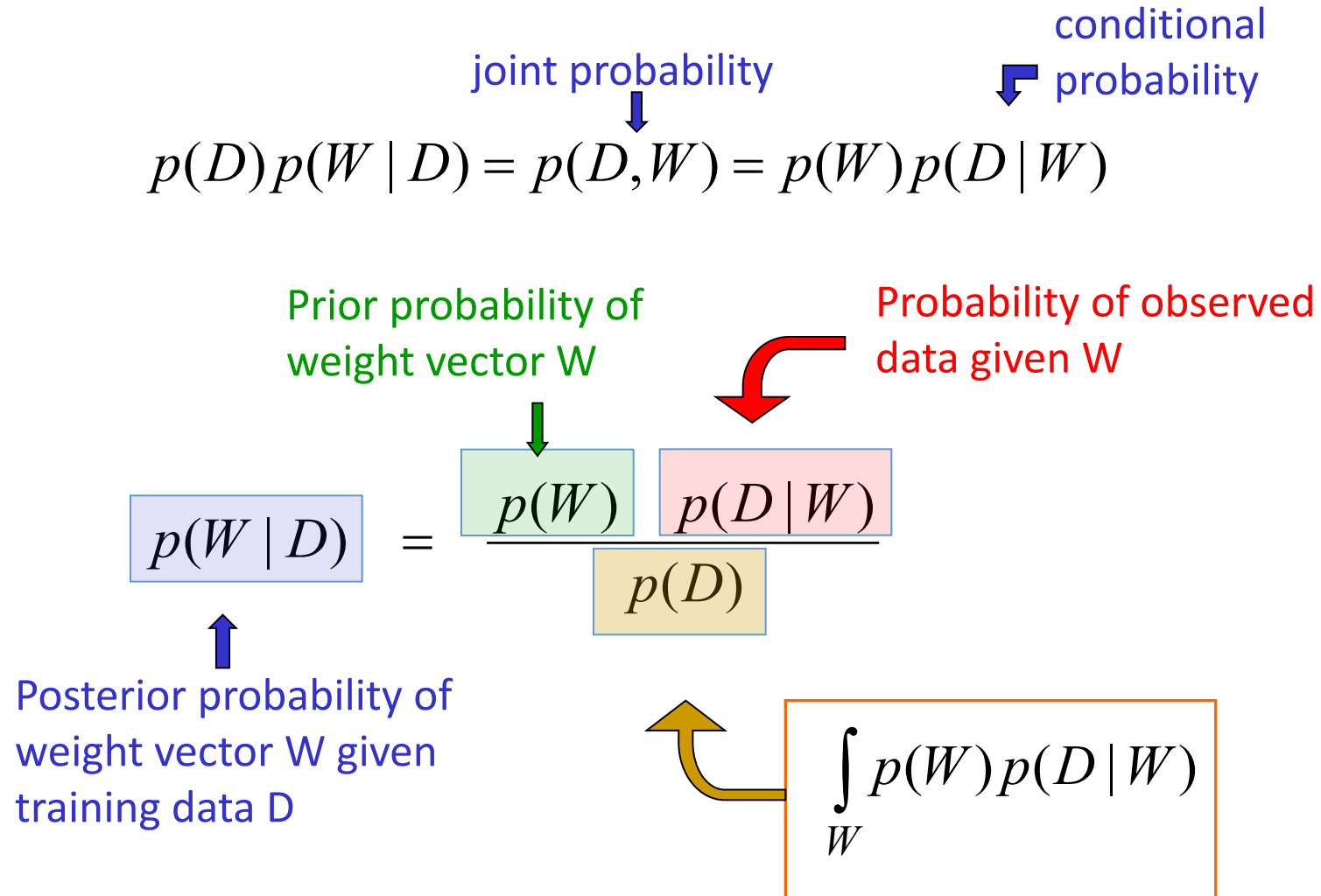
# Using a validation set

- Divide the total dataset into three subsets:
  - **Training data** is used for learning the parameters of the model.
  - **Validation data** is not used of learning but is used for deciding what type of model and what amount of regularization works best.
  - **Test data** is used to get a final, unbiased estimate of how well the network works. We expect this estimate to be worse than on the validation data.
- We could then re-divide the total dataset to get another unbiased estimate of the true error rate. (cross-validation)

# The Bayesian framework

- The Bayesian framework assumes that we always have a prior distribution for everything.
  - The prior may be very vague.
  - When we see some data, we combine our prior distribution with a likelihood term to get a posterior distribution.
  - The likelihood term takes into account how probable the observed data is given the parameters of the model.
    - It favors parameter settings that make the data likely.
    - It fights the prior
    - With enough data the likelihood terms always win.

# Bayes Theorem



# Maximize sums of log probs

- We want to maximize the **product** of the probabilities of the outputs on the training cases
  - Assume the output errors on different training cases,  $c$ , are independent.

$$p(D | W) = \prod_c p(d_c | W)$$

- Because the log function is monotonic, it does not change where the maxima are. So we can maximize **sums** of log probabilities

$$\log p(D | W) = \sum_c \log p(d_c | W)$$

- This is called **maximum likelihood** learning. It is very widely used for fitting models in machine learning.