CSCI-P556 **Applied Machine Learning**

Introduction

Luddy School, Indiana University

08/23&25/2021

Instructor: **Xuhong Zhang**

The slides throughout the semester were assembled by Xuhong Zhang, with grateful acknowledgement of the many others who made their course materials freely available online.

Today's Agenda

- Introduction: what is this class about
- Administrative: resources, grading etc.
- Machine Learning set up

Today's Agenda

- Introduction: what is this class about
- Administrative: resources, grading etc.
- Machine Learning set up

What is this class about?

- ***Basic theory*** and ***implementation*** of state-of-the-art machine learning algorithms for large-scale real-world applications.
- Topics include supervised learning (regression, classification, kernel methods, etc.) and unsupervised learning (clustering, dimensionality-related topics, etc.)

What is Machine Learning?

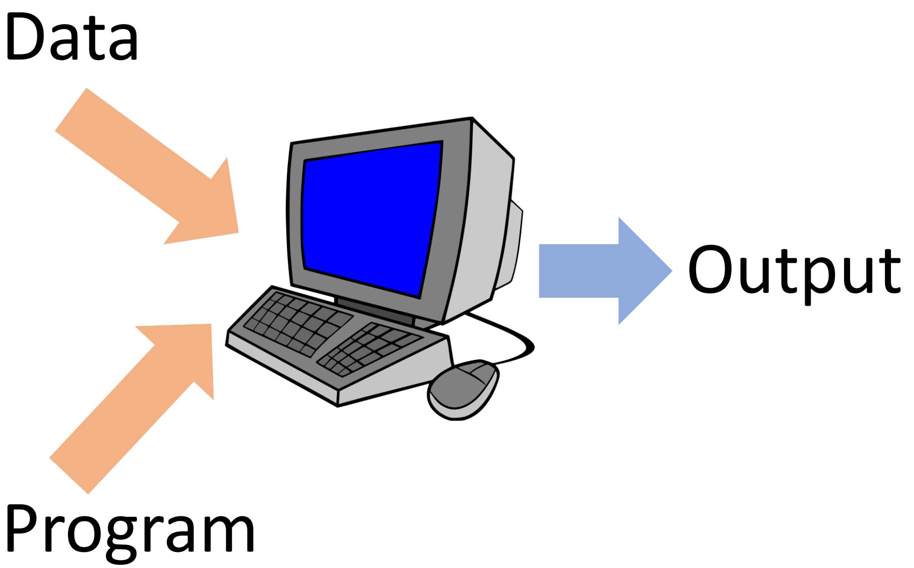
- “ Machine Learning is the study of **computer algorithms** that improve automatically through **experience** and by the **use of data**. It is seen as a part of artificial intelligence.”

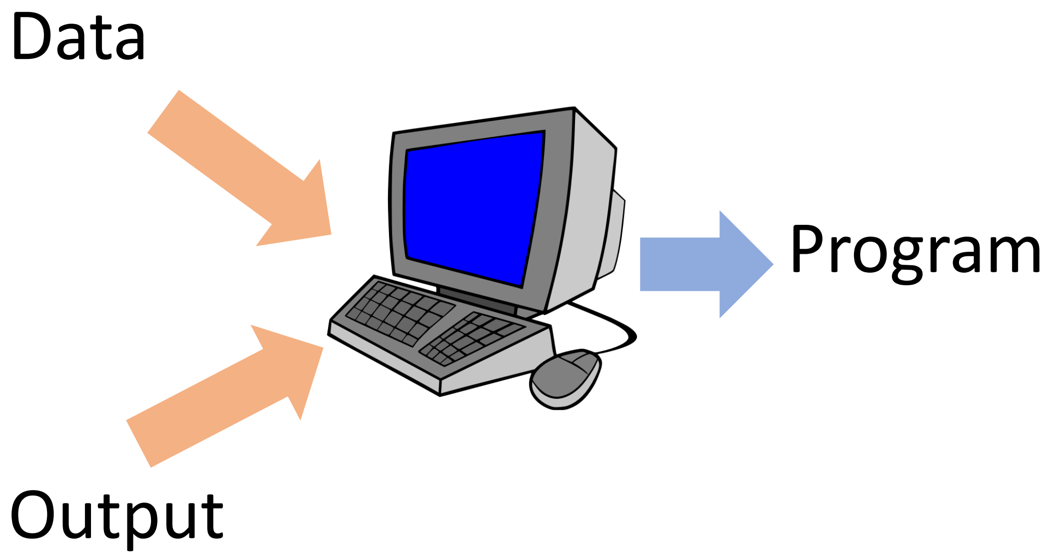
  -- Wikipedia

- Machine Learning is the study of algorithms that
  - Improve their performance *P*
  - At some task *T*
  - With experience *E*
  - A well-defined learning task is given by <*P*, *T*, *E*>

  -- Tom Mitchell (1998)

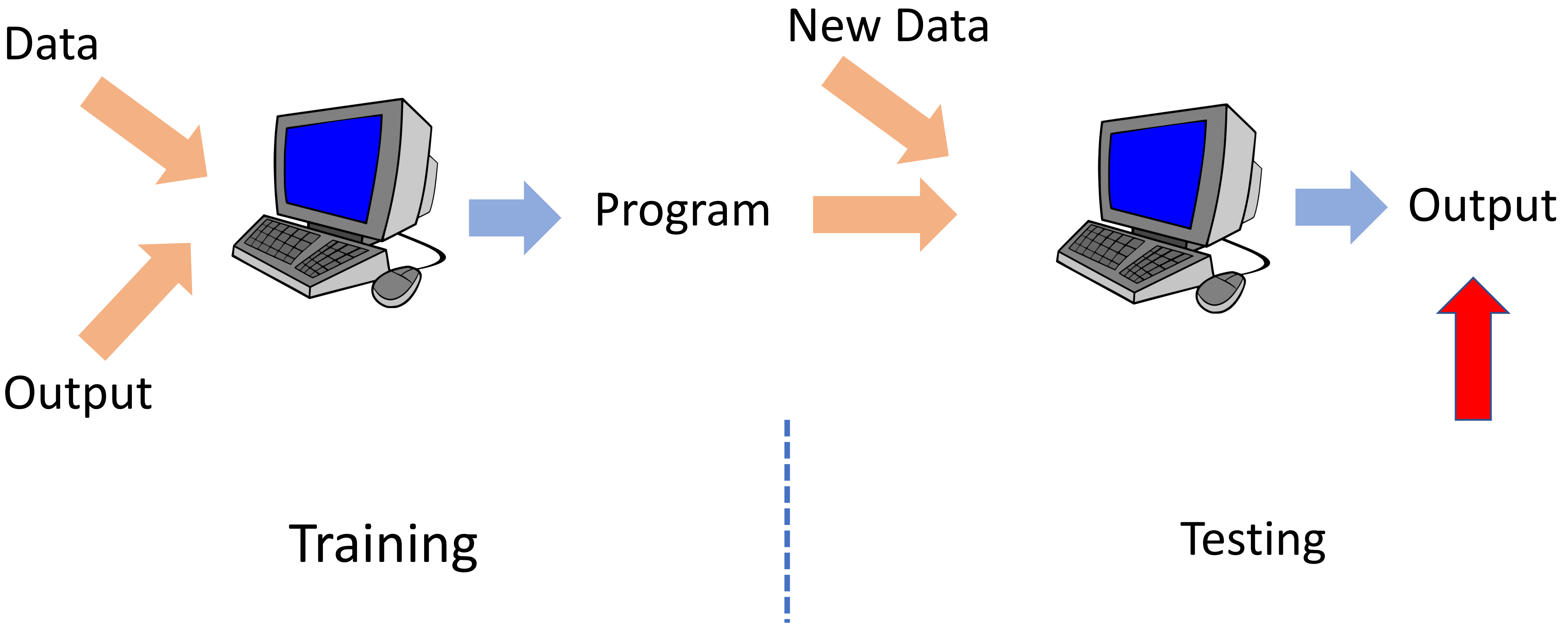
Traditional CS vs. Machine Learning





Traditional CS vs. Machine Learning

Machine Learning



Machine Learning vs. Statistics

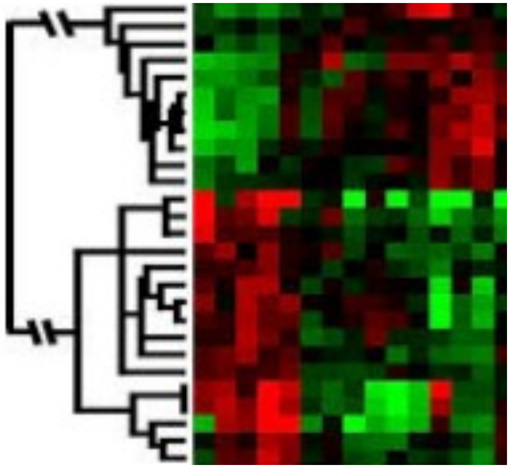
- Machine Learning
  - Data First / Data Driven
  - Prediction Emphasis
- Statistics
  - Model First / Model Driven
  - Inference Emphasis

When is Machine Learning needed?

When:

- Human expertise does not exist (navigating on Mars)
- It's hard to explain human's expertise (speech recognition, citation networks)
- Models must be customized (precision medicine)
- Models are based on huge amounts of data (genomics study)



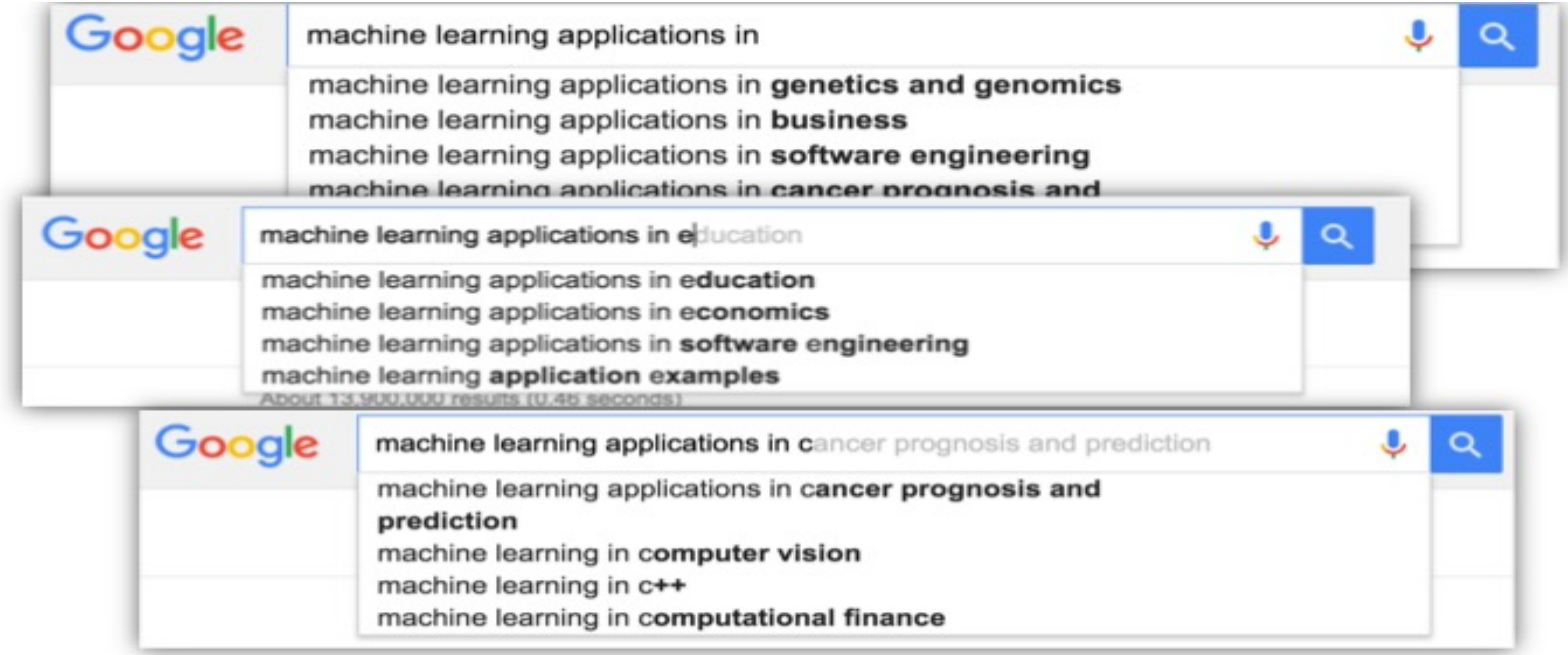



When Machine Learning is needed

- Learning isn't always useful:

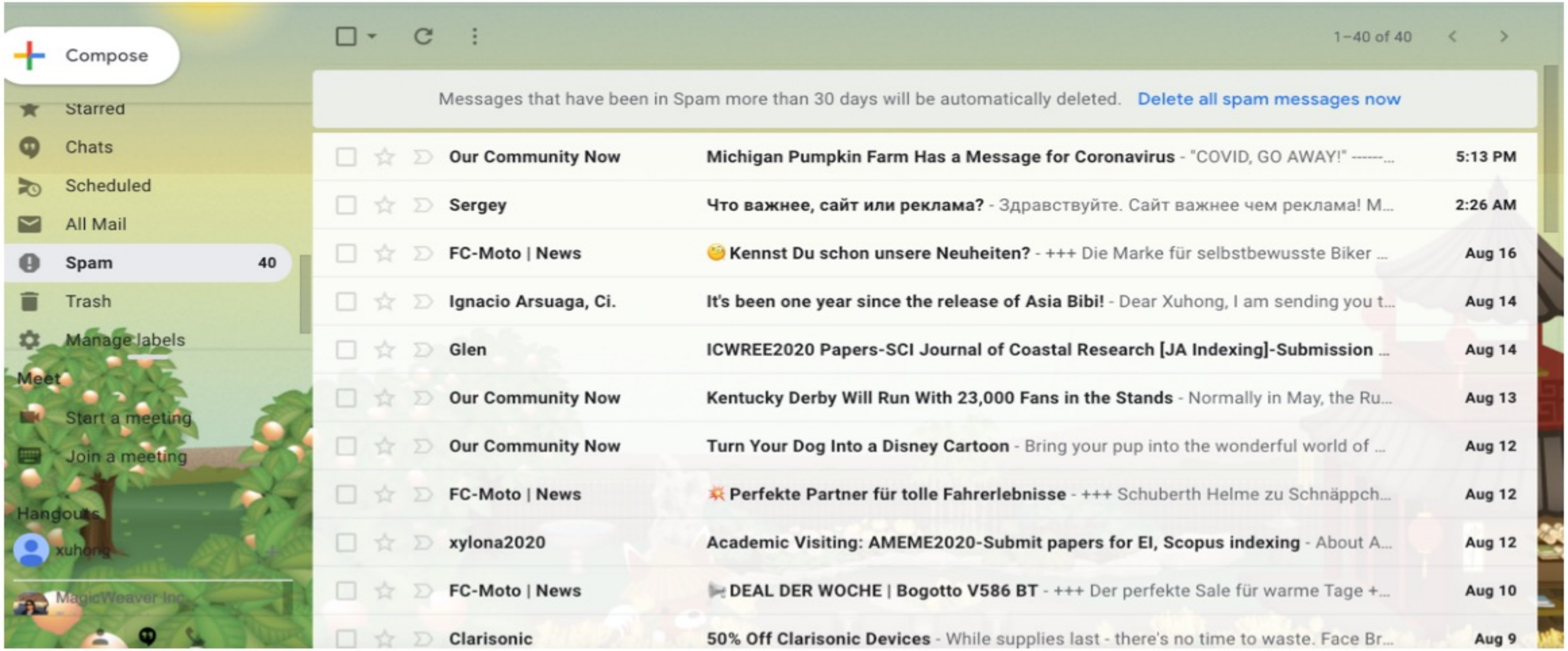
➢There is no need to "learn" to calculate payroll

Applications of Machine Learning



Classic examples of Machine Learning

- Spam Filter



Classic examples of Machine Learning

- Face Detection

More examples

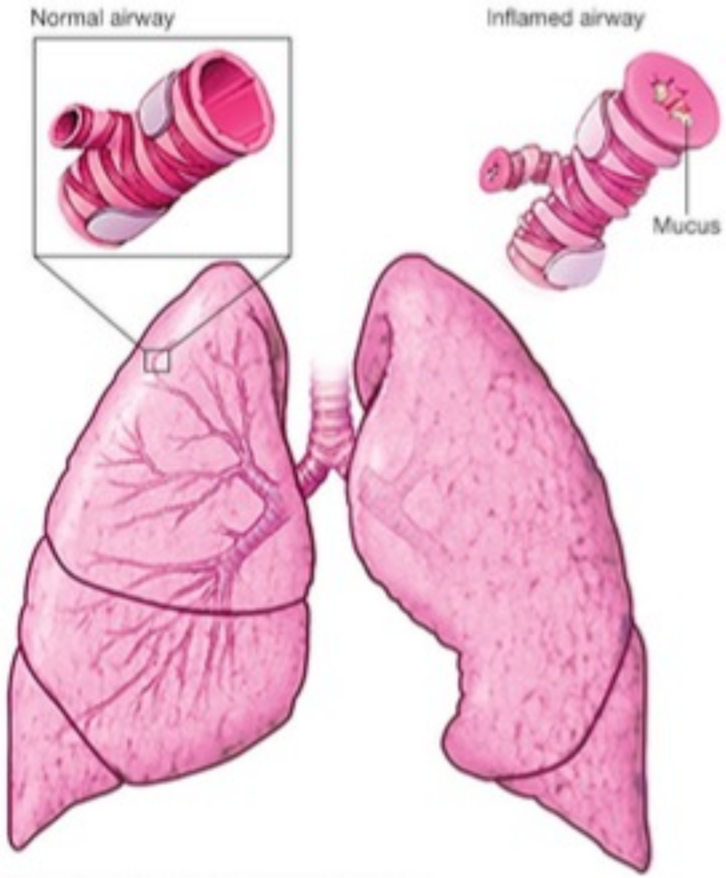
- Pattern Recognition:
  - Handwritten or spoken words
  - Medical images
- Pattern Generation:
  - Generating images or motion sequences
- Recognizing anomalies:
  - Unusual credit card transactions
  - Unusual patterns of sensor readings of automatic driving
- Prediction:
  - Future stock prices or housing prices

Use with caution !

"panda"
57.7% confidence

"gibbon"
99.3% confidence

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. and Fergus, R., 2013. Intriguing properties of neural networks.



“has Asthma(x)⇒Lower risk(x)”

Trade-off between interpretability and accuracy

Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-day Readmission
Caruana et. al. 2015

About this course

- The goal of this course it to help you understand the fundamentals of machine learning

- Provide foundations of machine learning
  - Basic mathematical derivation and implementation

- Cover practical applications of machine learning
  - Use machine learning algorithms for your problems/applications of interest

This is most important !

What this course *is not*

- Focused only on ***applied*** machine learning
  - we are interested in the basic mathematical interpretation of the algorithms
  - be prepared for "some math"

- Focused only on ***theoretical*** machine learning
  - we are also interested in applying algorithms to datasets to get hands-on experience with the algorithms
  - be prepared for some programming-heavy assignments

Today's Agenda

- Introduction: what is this class about
- Administrative: resources, grading etc.
- Machine Learning set up

Logistics

- Course Instructor: Xuhong Zhang (zhangxuh@iu.edu)
- Time : M & W, 7:00 PM – 8:15 PM
- Location : IF 0117
- Office : Luddy Hall, 3012
- Office Hours: Tuesday 9am-11am

Logistics

- **Pre-REQUISITE**
  - At least one-year experience of intensive programming
  - Questions you should not ask; Questions you can ask our AIs
- **Canvas**
  - Course syllabus, in-class quiz, slides, announcements, assignments, etc.
- **Tophat**
  - Interactions
- **Homework submission**
  - Course GitHub (Our Ais will send out more details regarding this.)
- **Piazza**
  - Discussion

Logistics

- Final Grade
  - Homework : 25 %
  - Bi-weekly In-Class Quiz : 20 % (First quiz starts Sep 1st—the forth lecture)
  - Final Exam : 30 %
  - Project: 25 % (Progress Report + Final Report + In-Class Presentation)
  - Course Evaluation: 1% (bonus)

- 4 homeworks (regular) + 1 bonus homework (deep learning)

- Late submission policy (see canvas)

Logistics

- Form your study group early on !
- For homework, you may discuss between the study group members, but you need to write your own solution **independently** ! (We have a tool to detect code copying and plagiarism)
- Please start on homework early (Warning: cramming does not work !)

Assignments: Homework

- There will be **4 regular homework** assignments and **1 bonus one**.
  - **Goal**: strengthen the understanding of the fundamental concept mathematical formulations, algorithms, and the applications.

  - The 1st homework will be due on Sep, 13th.
  - The 2nd homework will be due on Oct, 4th.
  - The 3rd homework will be due on Oct, 25th.
  - The 4th homework will be due on Nov, 15th.
  - The 5th (bonus) homework will be due on Dec, 13th.

Resources: Lectures

- Lecture slides and notes will be provided (Canvas)
- Optional readings will be assigned to complement lectures

Resources: Piazza

- Piazza can help you connect with other students in the class
- You can post questions
- You can answer each other's questions
- Assignment clarifications will be posted on piazza
- I and our AIs will review at regular intervals, but for a more immediate response come to office hours

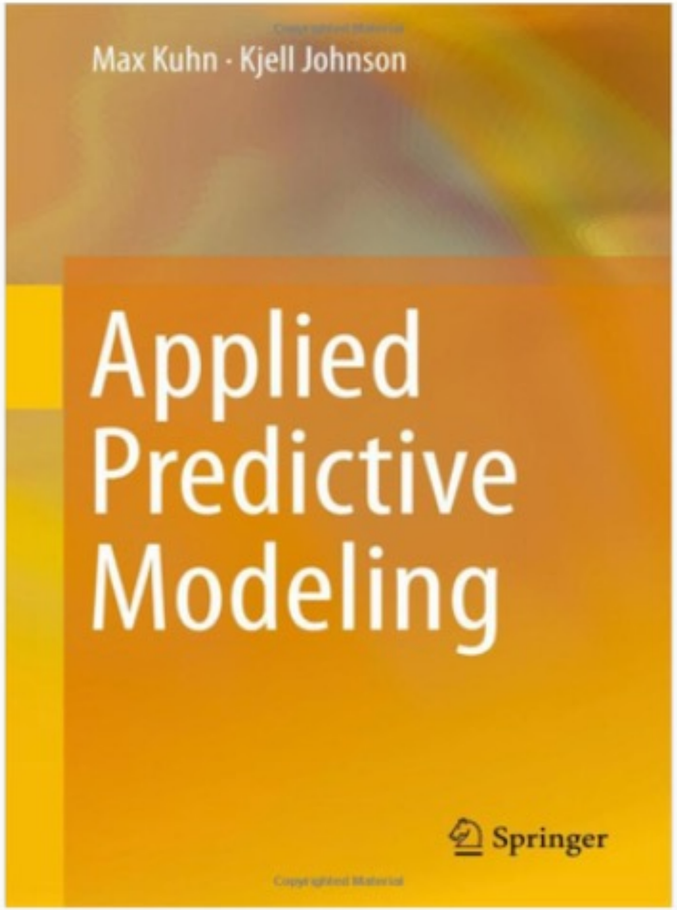
Code Copying and Plagiarism

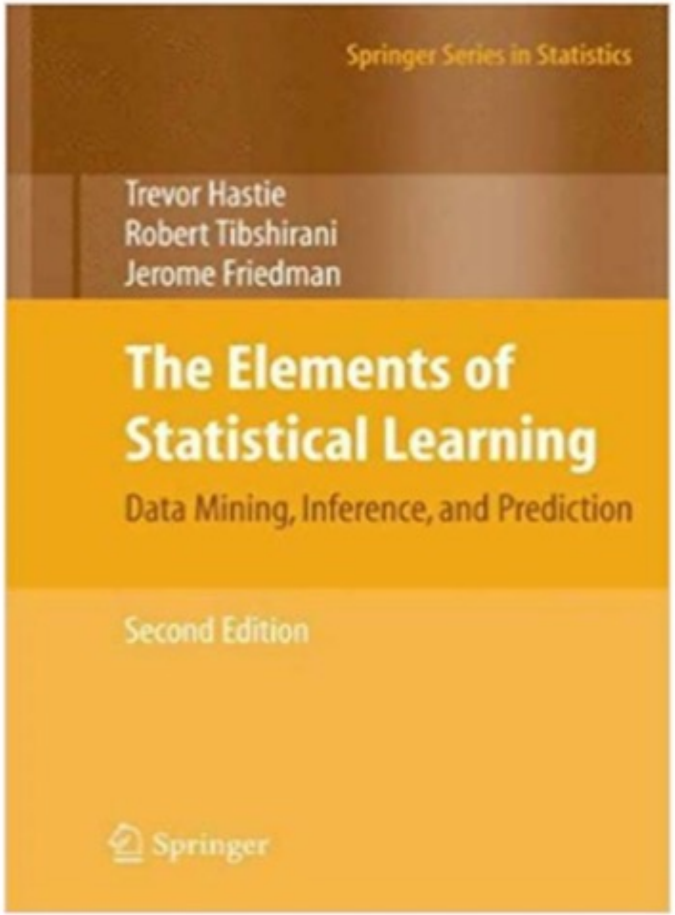
- Copied code will get 0 point for all involved
- Homework will be checked for plagiarism
- Copying from course code is fine
- Copying from online sources (stack overflow, tutorials, etc.) is fine but you have to refer to the source
- You also have to mention your member if you discuss together
- Plagiarism is not allowed throughout the entire semester

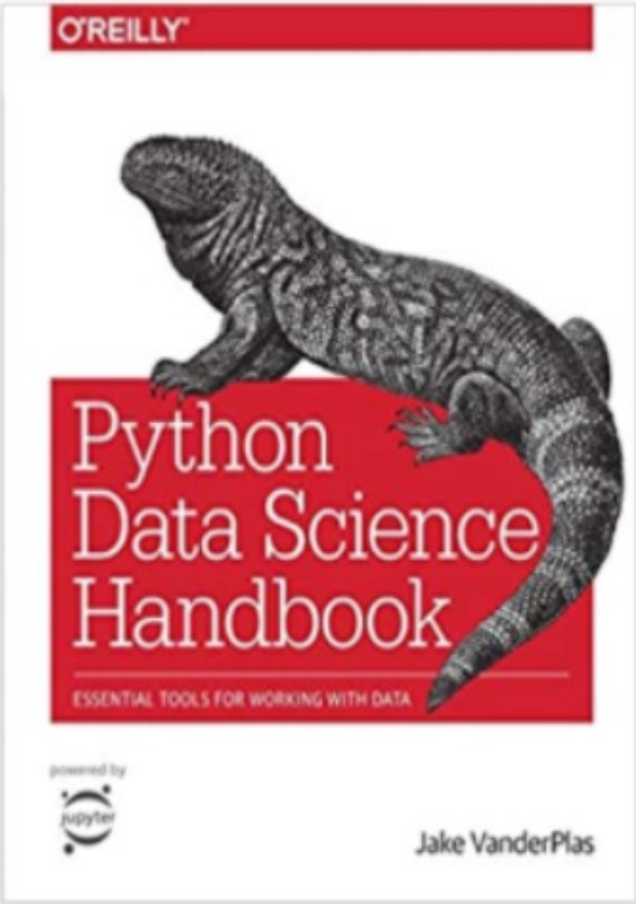
Programming Languages

- Python 3
  - Scikit-learn
  - Numpy
  - Scipy
  - Pandas
  - Matplotlib
  - …

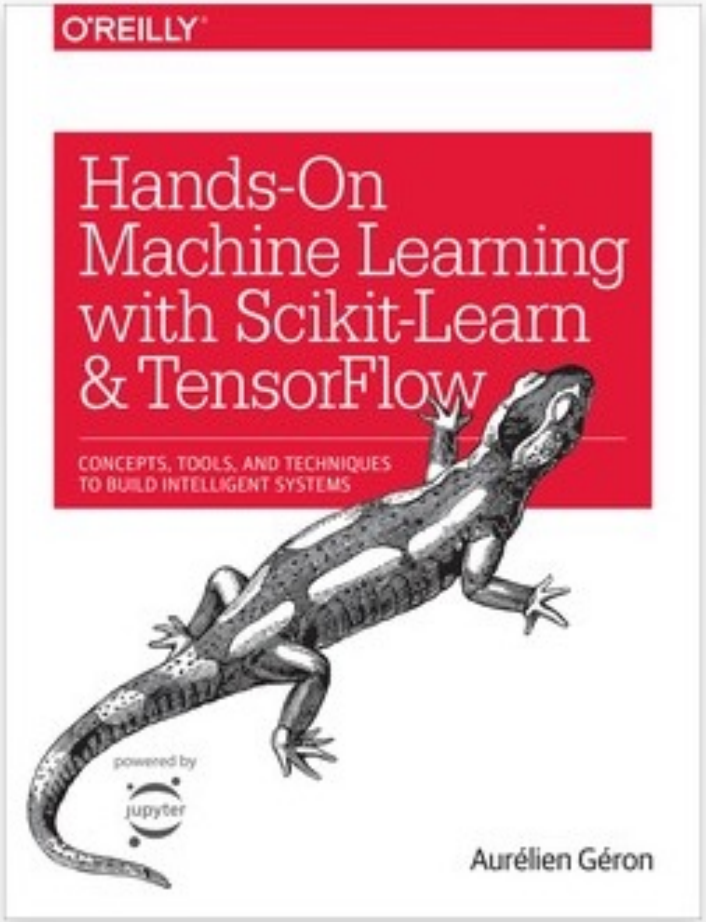
Books


Max Kuhn · Kjell Johnson
Applied Predictive Modeling
Springer


Springer Series in Statistics
Trevor Hastie
Robert Tibshirani
Jerome Friedman
The Elements of Statistical Learning
Data Mining, Inference, and Prediction
Second Edition
Springer


O'REILLY
Python Data Science Handbook
ESSENTIAL TOOLS FOR WORKING WITH DATA
powered by
jupyter
Jake VanderPlas

Books (optional)



Books (optional)

- A course in Machine Learning by Hal Daume III (available online)
- Pattern Recognition and Machine Learning by Christopher Bishop (available online)
- Mining of Massive Datasets by Leskovec, Rajaraman and Ullman (available online)
- Reinforcement Learning: An Introduction by Sutton and Barto (available online)

Tentative Schedule

- Due to the current situation, there may be unseen events during this semester, and the course schedule (topics not meeting time) might have to change accordingly

- Holidays (No class):
    - Sep, 6th (Labor day)
    - Nov, 22th, 24th (Thanks Giving)

Important Dates

- Will publish via Canvas
- Due to the current situation, there may be unseen events during this semester, and the course schedule might have to change accordingly
- Tentative dates:
  - Register your group before Sep, 22nd. (Will send out a google doc to put your group on it.)
  - Progress report due on Oct, 25th.
  - Final report due on Dec, 15th.

Previous Projects (20’ Fall)

- Hashtag Generator
- DNA and Protein Embedding
- Image Classification for Insufficient Datasets
- Spam/Ham Classification of Emails in English and Korean
- Food Item Recognition using CNN
- Real Time Object Recognition

Previous Projects (20’ Fall)

- A Stacking Method for Cancer Survival Classification
- Predicting the Recovery Time of Hospitalized Covid-19 Patients

Today's Agenda

- Introduction: what is this class about
- Administrative: resources, grading etc.
- Machine Learning set up

Defining the Learning Task

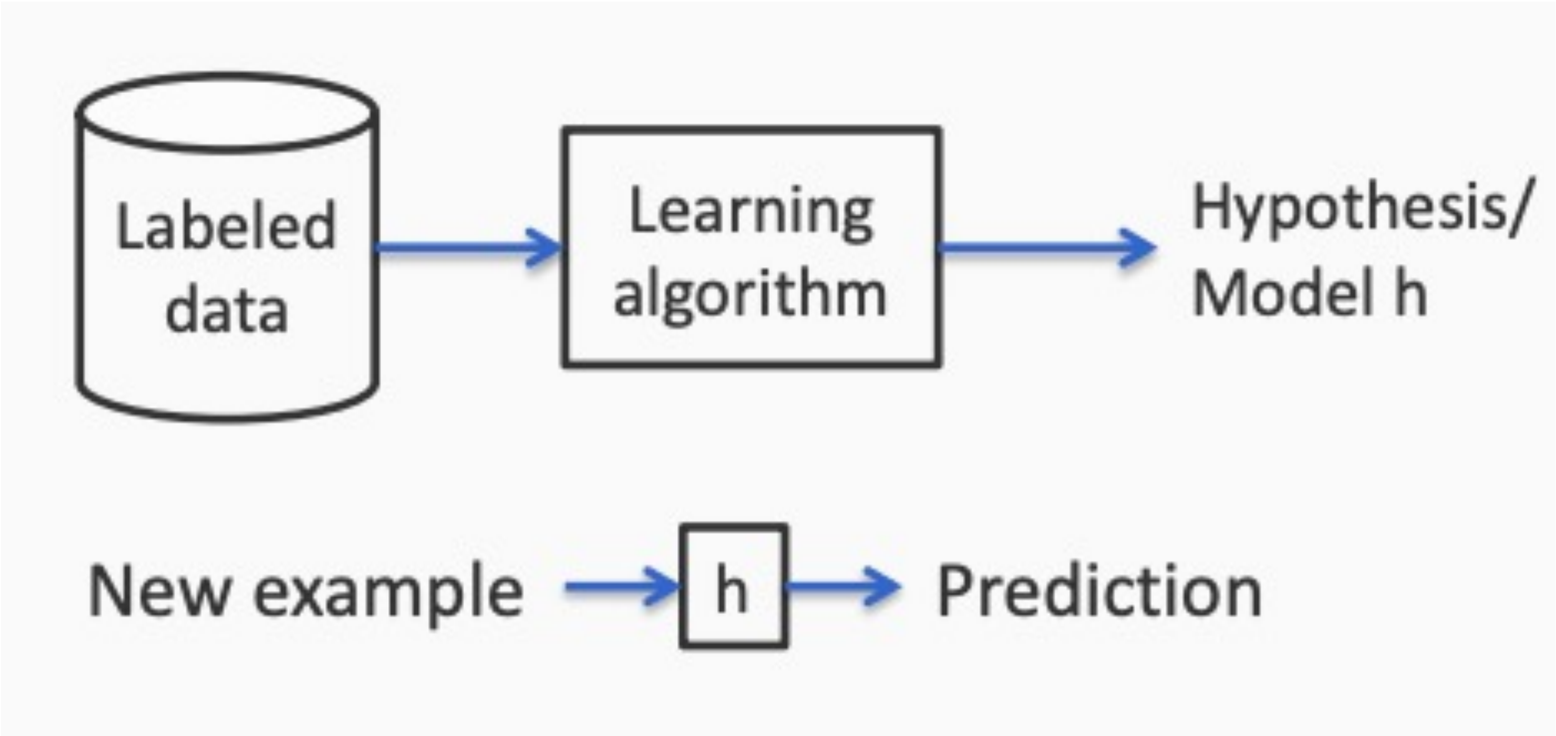
- Improve on task ***T***, with respect to performance metric ***P***, based on experience ***E***
  - ***T***: Categorize email messages as spam or legitimate
  - ***P***: Percentage of email messages correctly classified
  - ***E***: Database of emails, some with human-given labels

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

Types of Machine Learning

- *Supervised (inductive) Learning*
  - Given: training data + desired outputs (labels)
- *Unsupervised Learning*
  - Given: training data (without desired outputs)
- *Semi-supervised Learning*
  - Given: training data + a few desired outputs
- *Reinforcement Learning*
  - Rewards from sequence of actions

Supervised Learning



Supervised Learning

- Given input-output pairs, learn a function $f(x)$
  - $D = \{(x_i, y_i)_{i=1}^{N},\ (x_i, y_i) \propto p(x, y)\}, iid$
  - $f(x_i) \approx\ y_i$
  - $x_i \in\ \mathbb{R}^d$
  - y_i : categorical---classification
  - y_i : real valued---regression

Supervised Learning

- **Classification**

$$f(x_i) \approx y_i, y \in \{1, \dots, C\}$$

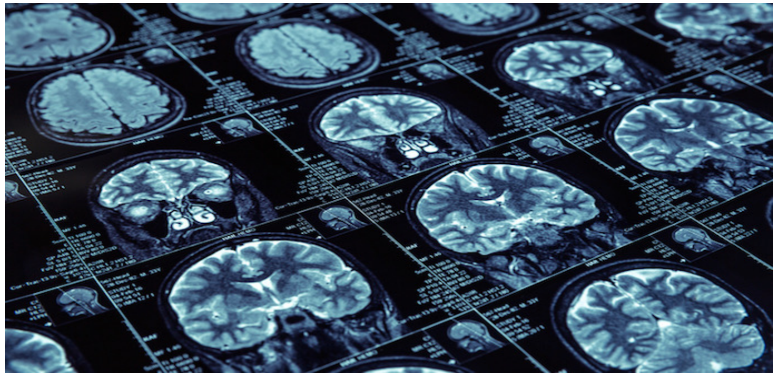
  - $C = 2$: binary classification
  - $C > 2$: multiclass classification

- **Regression**

$f(x_i) \approx y_i$, where y is continuous

Supervised Learning Examples

- Medical Image Learning

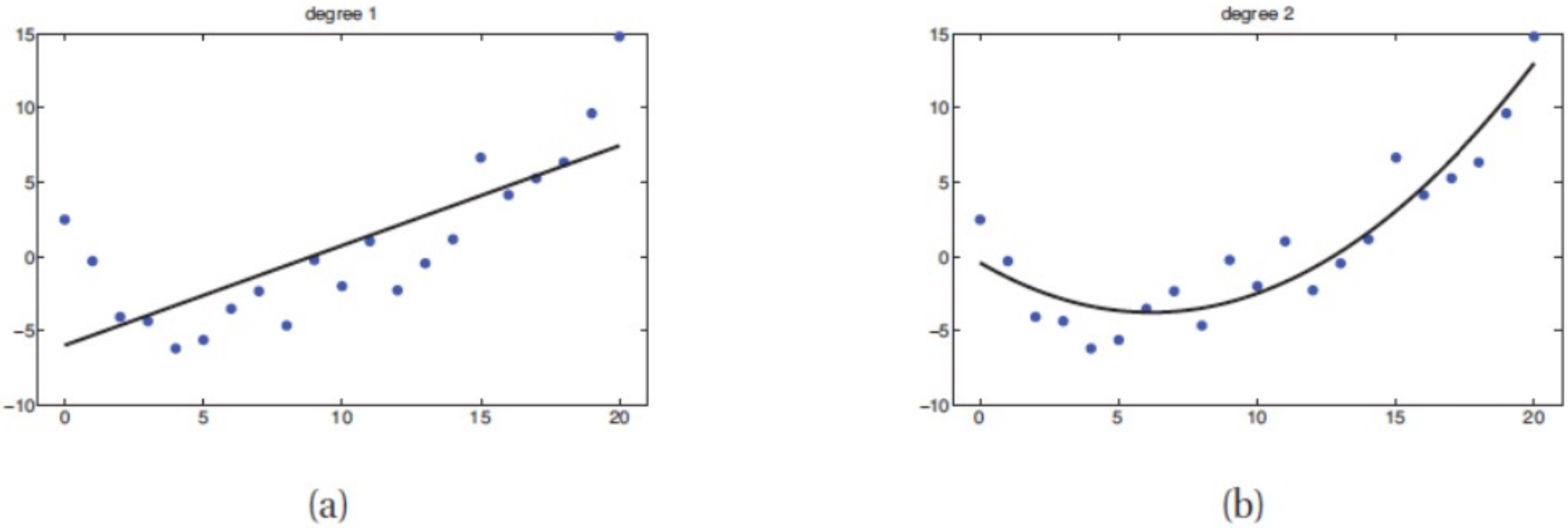


- Type Prediction

(a) (b) (c)

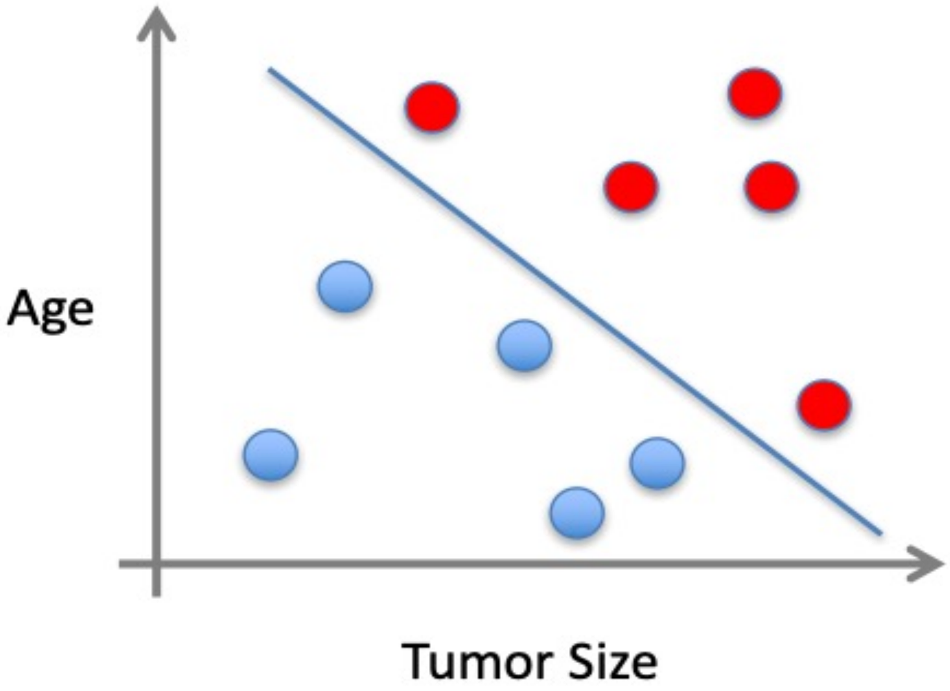
Supervised Learning Examples

- Regression



Supervised Learning Examples

- x can be multi-dimensional
  - Each dimension corresponds an attribute/feature/covariate



- Clump Thickness
- Uniformity of Cell Size
- Uniformity of Cell Shape

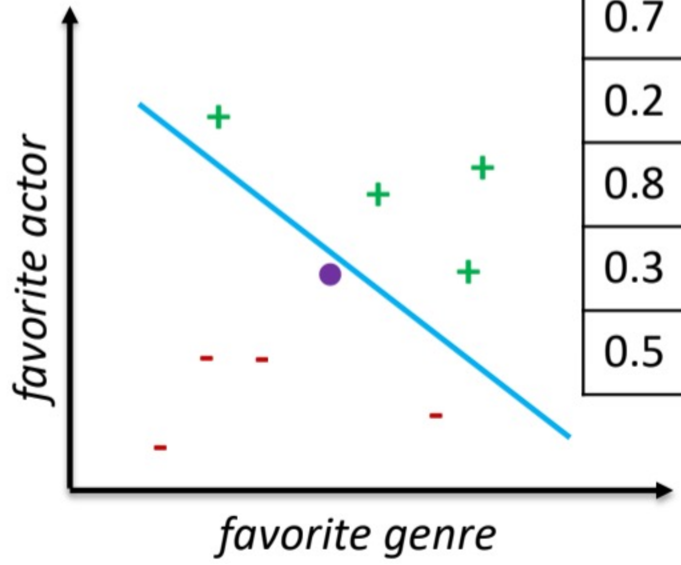
...

Based on example by Andrew Ng

Supervised Learning Examples

- Problem: predict whether a target user likes a target movie

- Data:
  - Features: percentage of your favorite genre scenes, percentage of scenes where your favorite actor appears
  - Labels: like/dislike



0.7	0.7	Like
0.2	0.2	Dislike
0.8	0.2	Dislike
0.3	0.9	Like
0.5	0.4	?

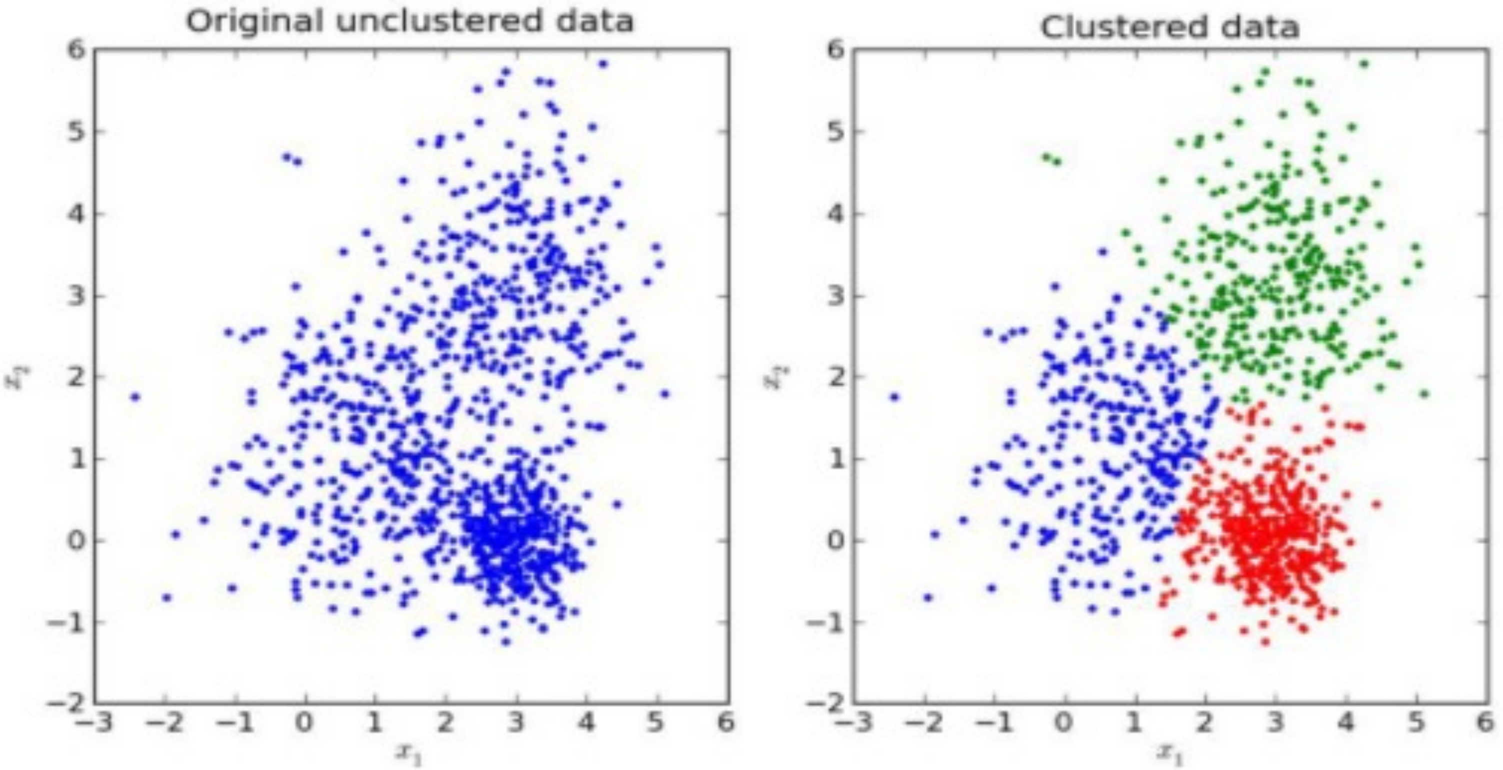
Goal: Learn a linear boundary

Unsupervised Learning

- Input Data
  - $D = \{x_i\}_{i=1}^{N}, \, x_i \propto p(x), \; iid$
  - Learn about P

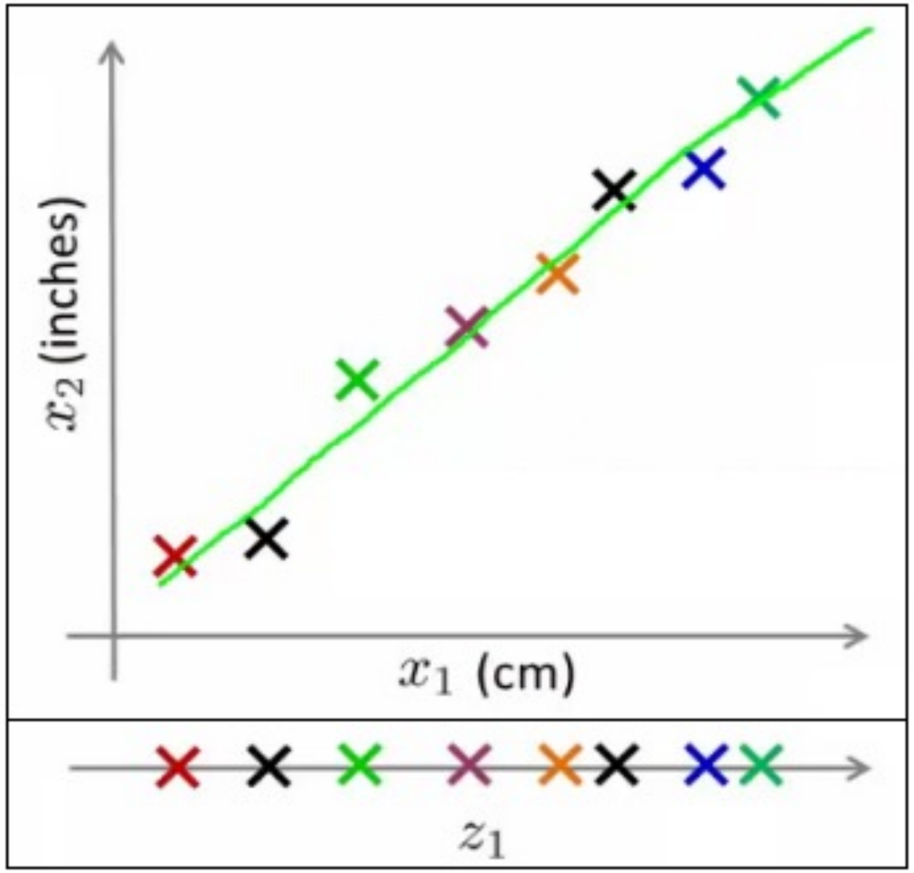
Unsupervised Learning Examples

- Clustering



Unsupervised Learning Examples

- Dimensionality Reduction



Unsupervised Learning Examples

- Topic Modeling

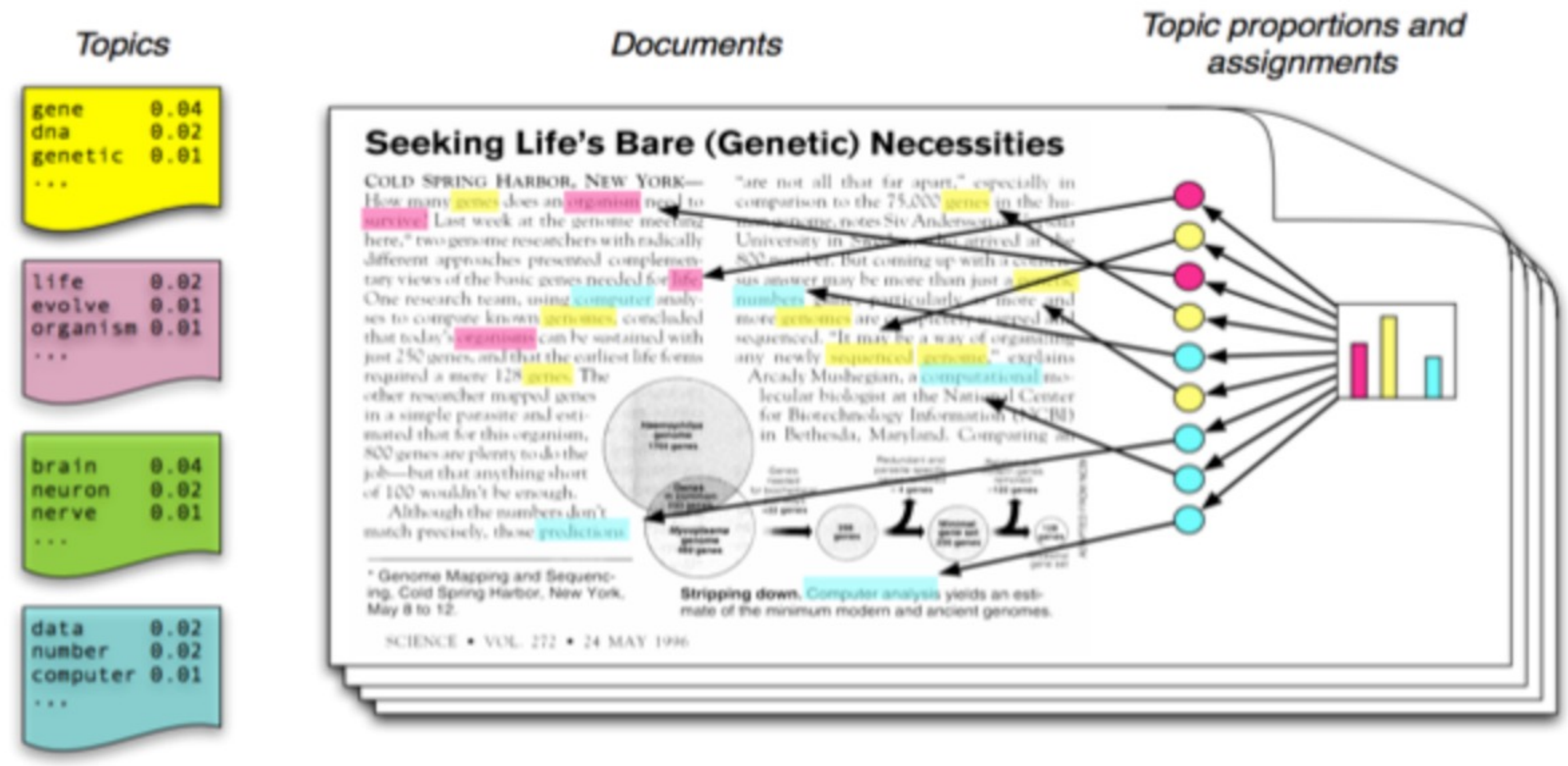
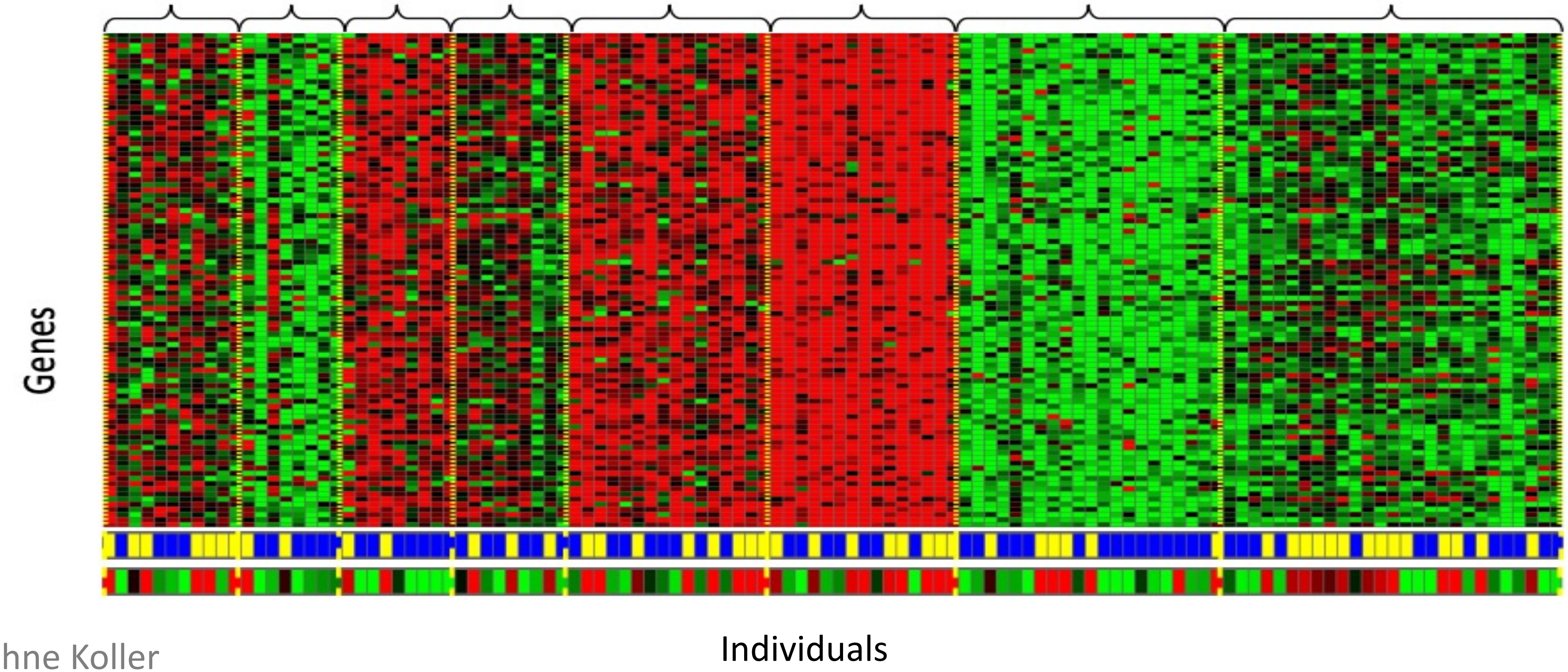


Figure source: Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, *55*(4), 77-84.

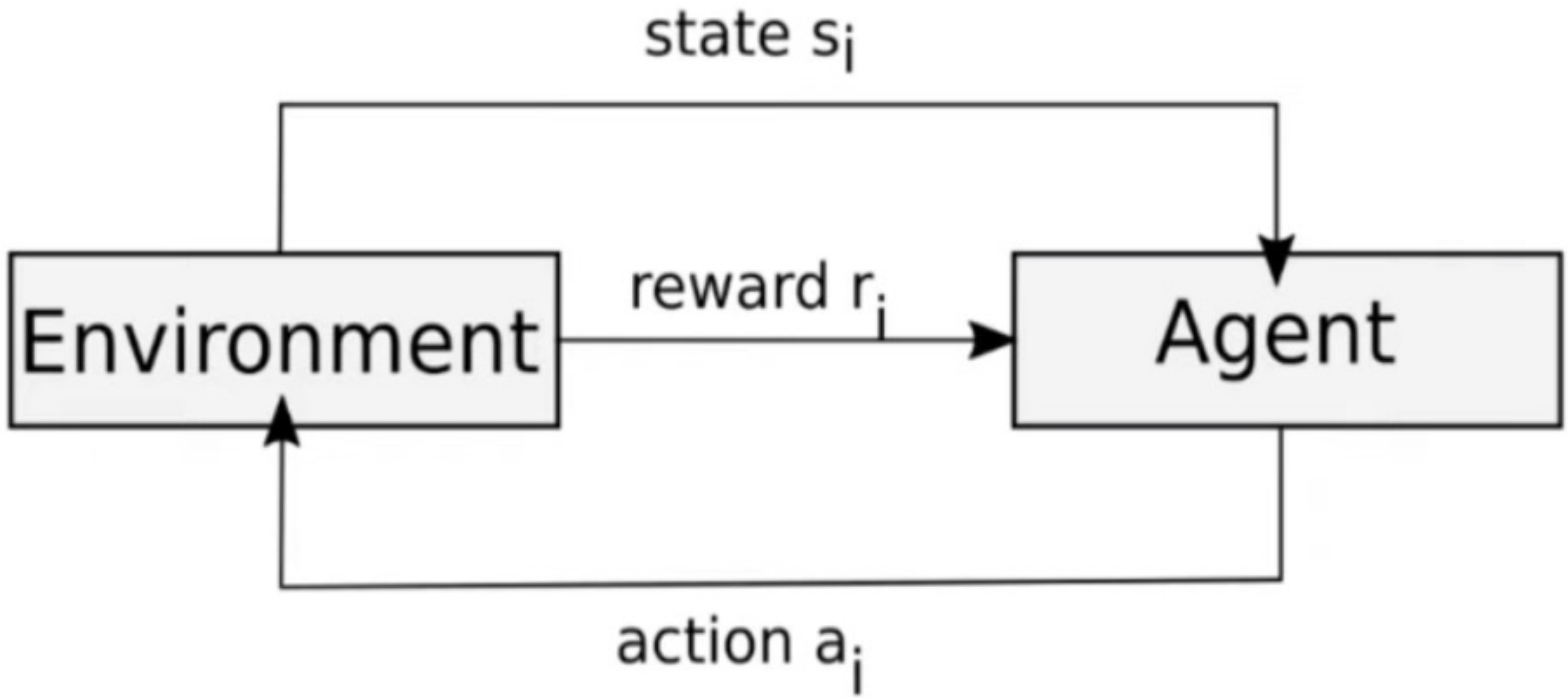
Unsupervised Learning Examples

- Genomics application: group individuals by genetic similarity



Source: Daphne Koller

Reinforcement Learning



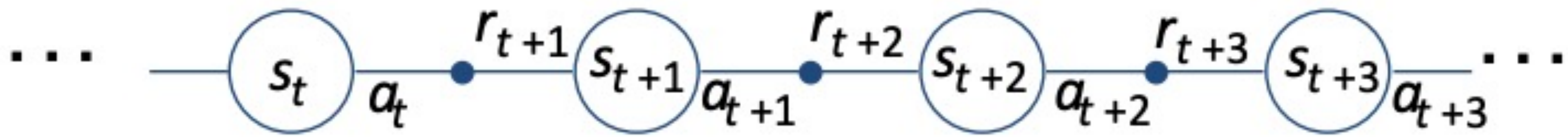
Reinforcement Learning

- Given a sequence of stats and actions with (delayed) rewards, output a policy
  - Policy is a mapping from states to actions that tells you what to do in a given state

- Examples
  - Game playing
  - Robot in a maze

Reinforcement Learning

- Agent and environment interact a discrete time steps: $t = 0, 1, \dots, K$
  - Agent observes state at step t: $S_t \in S$
  - Produces action at step t: $a_t \in A(S_t)$
  - Get resulting reward: $r_{t+1} \in \Re$
  - And resulting next state: S_{t+1}



Slide credit: Sutton & Barto

Reinforcement Learning Examples

- Alpha Go

Reinforcement Learning Examples

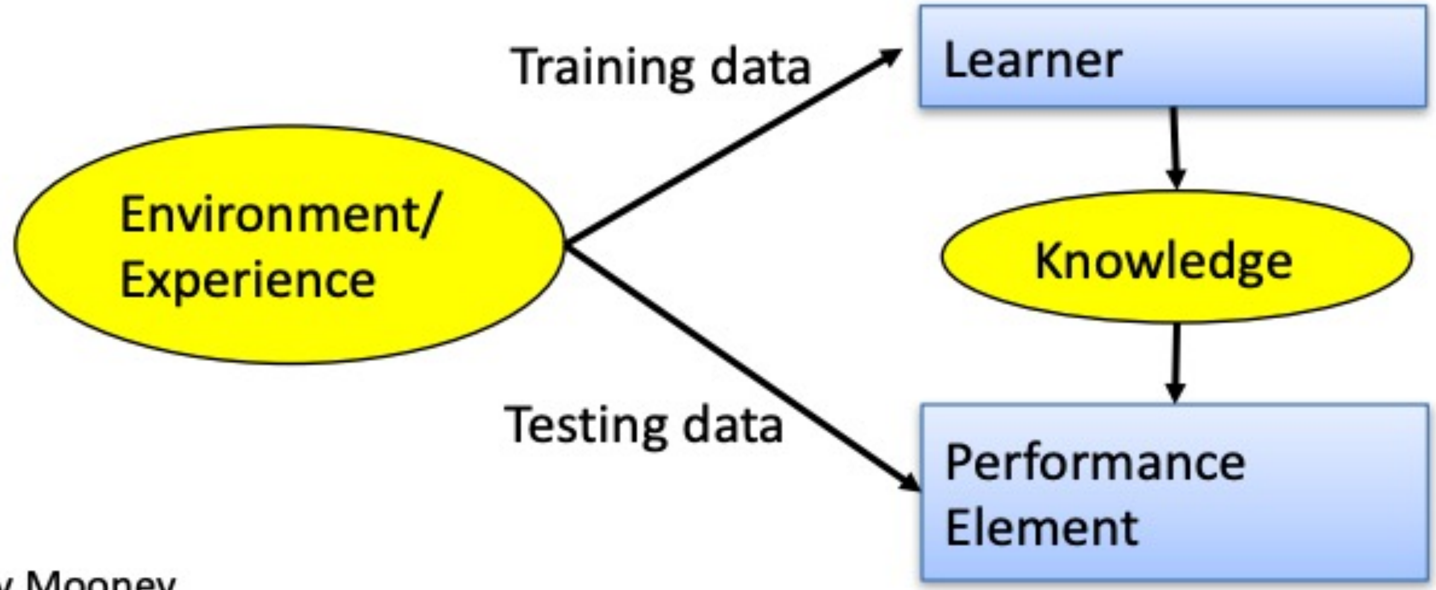
- Self-Driving Car

Other Types

- Semi-supervised
- Active Learning
- Forecasting
- …

How to frame a learning task

- 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 experience



Based on slide by Ray Mooney

Training vs. Test Distribution

- We generally assume that the training and test examples are independently drawn from the same overall distribution of data
  - We call this "i.i.d" which stands for "independent and identically distributed"
- If examples are not independent, requires *collective classification*
- If test distribution is different, requires *transfer learning*

ML in a Nutshell

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

Slide Credit: Pedro Domingos

Various Function Representations

- Numerical functions
  - *Linear regression*
  - *Neural networks*
  - *Support vector machines*
- Symbolic functions
  - *Decision trees*
  - *Rules in propositional logic*
  - *Rules in first-order predicate logic*
- Instance-based functions
  - *Nearest-neighbor*
  - *Case-based*
- Probabilistic Graphical Models
  - *Naïve Bayes*
  - *Bayesian networks*
  - *Hidden-Markov Models (HMMs)*
  - *Probabilistic Context Free Grammars*
  - *Markov networks*

Slide credit: Ray Mooney

Various Search/Optimization Algorithms

- Gradient descent
  - *Perceptron*
  - *Backpropagation*
- Dynamic Programming
  - *HMM Learning*
  - *PCFG Learning*
- Divide and Conquer
  - *Decision tree induction*
  - *Rule learning*
- Evolutionary Computation
  - *Genetic Algorithms (GAs)*
  - *Genetic Programming (GP)*
  - *Neuro-evolution*

Slide credit: Ray Mooney

ML in Practice

Loop

- Understand domain, prior knowledge, and goals
- Data integration, selection, cleaning, pre-processing, etc
- Learn models
- Interpret results
- Consolidate and deploy discovered knowledge

Slide credit: Pedro Domingos

Lessons learned about learning

- Learning can be viewed as using direct or indirect experience to approximate a chosen target function.
- Function approximation can be viewed as a search through a space of hypotheses (representations of functions) for one that best fits a set of training data.
- Different learning methods assume different hypothesis spaces (representation languages) and/or employ different search techniques.

Slide credit: Ray Mooney

History of Machine Learning

✓**1960s**
- ***Neural networks: Perceptron***
- Pattern recognition
- Learning in the limit theory

✓**1980s**
- ***Advanced decision tree and rule learning***
- Explanation-based learning (EBL)
- Learning and planning and problem solving
- Utility problem
- Analogy
- Resurgence of neural networks (connectionism, backpropagation)
- Valiant's PAC learning Theory

Slide credit: Ray Mooney

History of Machine Learning

✓**1990s**

- Data mining
- Adaptive software agents and web applications
- ***Text mining***
- ***Reinforcement Learning (RL)***
- Inductive Logic Programming (ILP)
- ***Ensembles: Bagging, Boosting, and Stacking***
- ***Bayes Net Learning***

Slide credit: Ray Mooney

History of Machine Learning

✓**2000s**

- ***Support vector machines & kernel methods***
- ***Graphical models***
- Statistical relational learning
- ***Transfer learning***
- Sequence labeling
- Collective classification and structured outputs
- Computer Systems Applications (Compilers, Debugging, Graphics, Security)
- E-mail management
- Personalized assistants
- ***Learning in robotics and vision***

Slide credit: Ray Mooney

History of Machine Learning

✓**2010s**

- ***Deep learning systems***
- ***Learning for big data***
- ***Bayesian methods***
- Multi-task & lifelong learning
- Applications to vision, speech, social networks, learning to read, etc.
- …

Slide credit: Ray Mooney

Sidebar: Ethical Considerations

- Privacy
- Fairness and bias
- Benefit vs. Harm
- …

What we'll cover in this course

❖Supervised Learning
- Distance based classification
- Linear regression
- Logistic regression
- Perceptron
- Support Vector Machines
- Ensembles
- Neural networks & Deep learning
- Trees

❖Unsupervised Learning
- Clustering
- Dimensionality reduction

❖Optimization methods

❖Model Evaluation

❖Applications

We will more focus on applying machine learning to real applications

Basic Concepts (1)

- **Parametric** vs. **non-parametric** models
    - Parametric: all the parameters are in finite-dimensional parameter spaces
    - Non-parametric: all the parameters are in infinite-dimensional parameter spaces. The model structure is not specified a priori but is instead determined from the data.

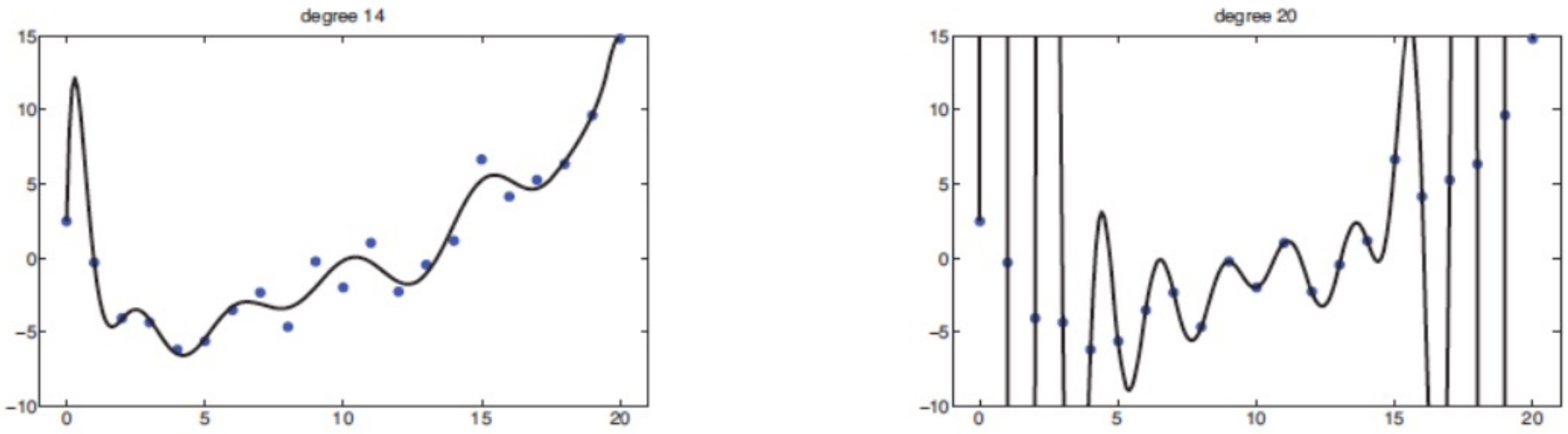
Basic Concepts (1)

- Parametric model examples
  - Exponential family
  - Poisson family
  - …

- Non-parametric model examples
  - K-nearest neighbor
  - Kernel density estimation
  - …

Basic Concepts (2)

- Overfitting

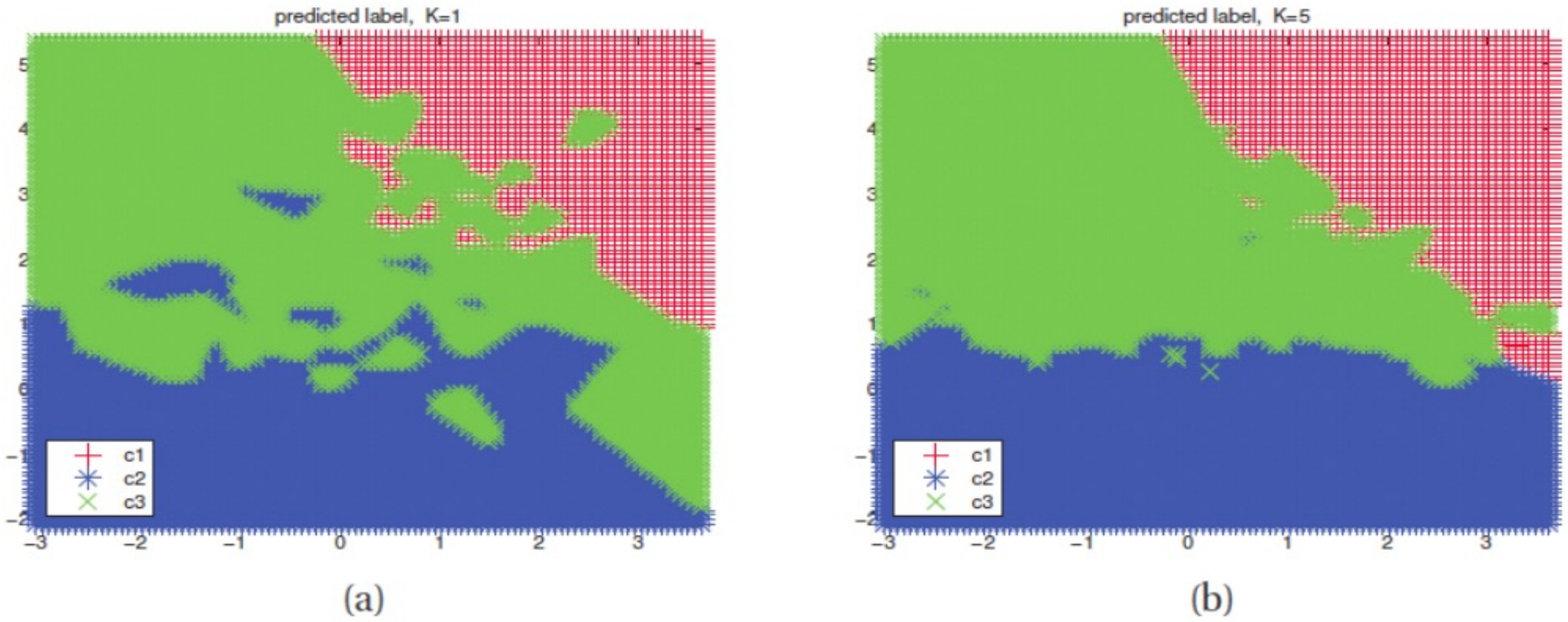


(a) (b)

Polynomial of degrees 14 and 20 fit by least squares to 21 data points. Figure generated by linewfPolyVsDegree in Matlab.

Basic Concepts (2)

- Overfitting



Prediction surface for KNN on the training data. (a) K = 1. (b) K = 5. Figure generated by knnClassifyDemo in Matlab.

Basic Concepts (2)

- Let $\mathcal{H}$ denotes the set of classifiers under consideration

- Too many choices not always a good thing
  - May lead to overfitting
- Solution?
  - Constrain possible choices, $\mathcal{H}$
- Caution !
  - $\mathcal{H}$ cannot be too constrained either

- This problem is called model selection

Basic Concepts (3)

- Generalization
  - For supervised learning, we not only learn

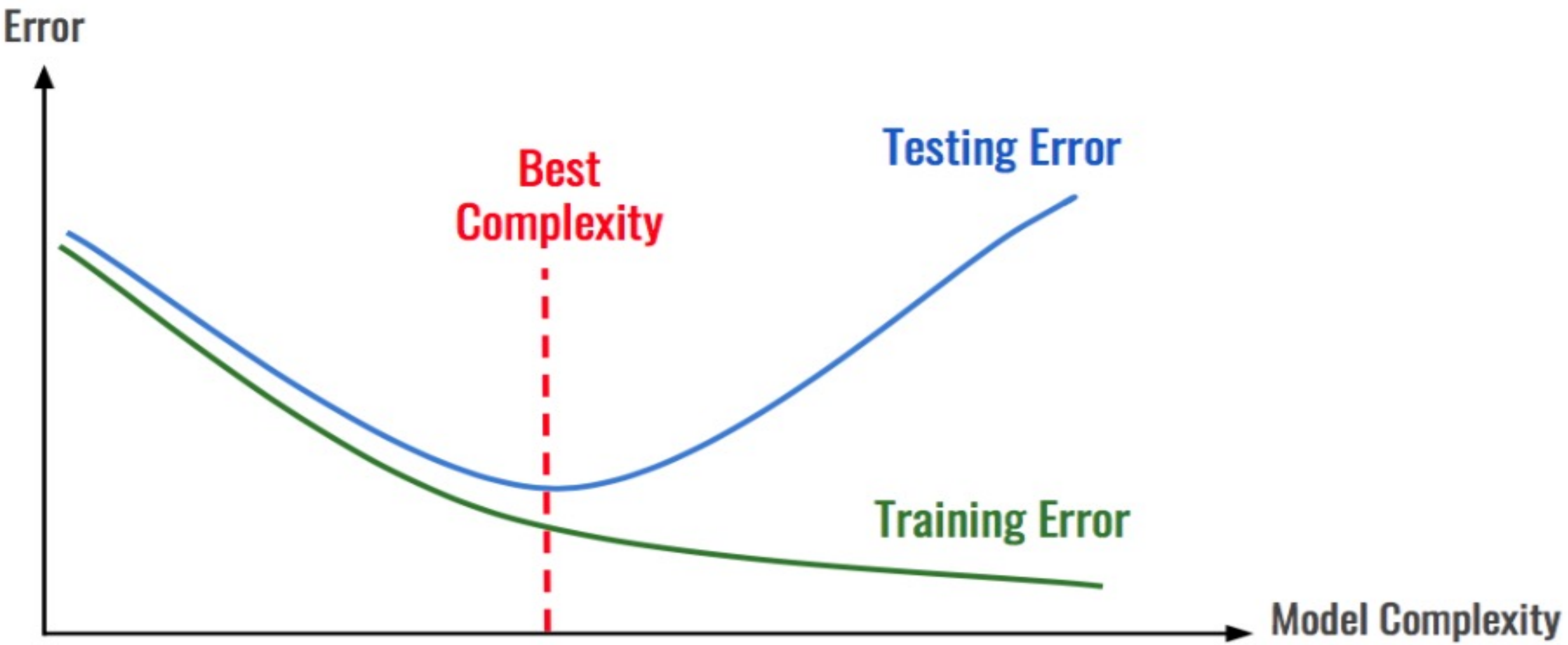
$$f(x_i) \approx y_i$$

  - More important, we want

$$f(x_{new}) \approx y_{true}$$

Basic Concepts (4)

- Model Selection



Knowing Your Goal and Your Data

- What question(s) am I trying to answer? Do I think the data collected can answer that question?

- What is the best way to phrase my questions(s)?

- Have I collected enough data to represent the problem I want to solve?

  - Plot your data !!

Knowing Your Goal and Your Data

- What features of the data did I extract, and will these enable the right predictions?
- How can I measure success in my application?
- Can I interpret the model and the process to someone else?