Machine Learning 10-401, Spring 2018

Introduction, Admin, Course Overview

Lecture 1, 01/17/2018

Maria-Florina (Nina) Balcan
Machine Learning

Image Classification

Document Categorization

Speech Recognition    Protein Classification    Spam Detection

Branch Prediction    Fraud Detection    Natural Language Processing

Playing Games    Computational Advertising
Machine Learning is Changing the World

“Machine learning is the hot new thing”
(John Hennessy, President, Stanford)

“A breakthrough in machine learning would be worth ten Microsofts” (Bill Gates, Microsoft)

“Web rankings today are mostly a matter of machine learning”
(Prabhakar Raghavan, VP Engineering at Google)
The COOLEST TOPIC IN SCIENCE

- “A breakthrough in machine learning would be worth ten Microsofts” (Bill Gates, Chairman, Microsoft)
- “Machine learning is the next Internet” (Tony Tether, Director, DARPA)
- Machine learning is the hot new thing” (John Hennessy, President, Stanford)
- “Web rankings today are mostly a matter of machine learning” (Prabhakar Raghavan, Dir. Research, Yahoo)
- “Machine learning is going to result in a real revolution” (Greg Papadopoulos, CTO, Sun)
- “Machine learning is today’s discontinuity” (Jerry Yang, CEO, Yahoo)
This course: introduction to machine learning.

• Cover (some of) the most commonly used machine learning paradigms and algorithms.
  • Sufficient amount of details on their mechanisms: explain why they work, not only how to use them.
  • Applications.
What is Machine Learning?

Examples of important machine learning paradigms.
Supervised Classification
from data to discrete classes
Supervised Classification. Example: Spam Detection

Decide which emails are spam and which are important.

Goal: use emails seen so far to produce good prediction rule for future data.
## Supervised Classification. Example: Spam Detection

Represent each message by features. (e.g., keywords, spelling, etc.)

<table>
<thead>
<tr>
<th>“money”</th>
<th>“pills”</th>
<th>“Mr.”</th>
<th>bad spelling</th>
<th>known-sender</th>
<th>spam?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

**Reasonable RULES:**

Predict SPAM if unknown AND (money OR pills)

Predict SPAM if 2money + 3pills −5 known > 0

Linearly separable
Supervised Classification. Example: Image classification

- Handwritten digit recognition (convert hand-written digits to characters 0..9)

- Face Detection and Recognition
Supervised Classification. Many other examples

- Weather prediction

- Medicine:
  - diagnose a disease
    - input: from symptoms, lab measurements, test results, DNA tests, …
    - output: one of set of possible diseases, or “none of the above”
    - examples: audiology, thyroid cancer, diabetes, …
      - or: response to chemo drug X
      - or: will patient be re-admitted soon?

- Computational Economics:
  - predict if a stock will rise or fall
  - predict if a user will click on an ad or not
    - in order to decide which ad to show
Regression. Predicting a numeric value

Stock market

Weather prediction

Predict the temperature at any given location
Other Machine Learning Paradigm

Clustering: discovering structure in data (only unlabelled data)

- E.g., cluster users of social networks by interest (community detection).

Semi-Supervised Learning: learning with labeled & unlabelled data

Active Learning: learns pick informative examples to be labeled

Reinforcement Learning (accommodates indirect or delayed feedback)

Dimensionality Reduction

Collaborative Filtering (Matrix Completion), …
Many communities relate to ML
Admin, Logistics, Grading
Brief Overview

• **Meeting Time**: Mon, Wed, NSH 3002, 10:30 – 11:50

• **Course Staff**
  
  • **Instructors**:
    
    – Maria Florina (Nina) Balcan (ninamf@cs.cmu.edu)
  
  • **TAs**:
    
    – Kenneth Marino (kdm Marino@cs.cmu.edu)
    – Colin White (crwhite@cs.cmu.edu)
    – Nupur Chatterji (nchatter@andrew.cmu.edu)
Brief Overview

• Course Website
  
  http://www.cs.cmu.edu/~ninamf/courses/401sp18

• See website for:
  • Syllabus details
    • All the lecture slides and homeworks
    • Additional useful resources.
  • Office hours
  • Recitation sessions
  • Grading policy
  • Honesty policy
  • Late homework policy
  • Piazza pointers

• Will use Piazza for discussions.
Prerequisites. What do you need to know now?

- You should know how to do math and how to program:
  - Calculus (multivariate)
  - Probability/statistics
  - Algorithms. Big O notation.
  - Linear algebra (matrices and vectors)
  - Programming:
    - You will implement some of the algorithms and apply them to datasets
    - Assignments will be in Octave (play with that now if you want; also recitation tomorrow)
    - Octave is open-source software clone of Matlab.
  - We may review these things but we will not teach them
Source Materials

No textbook required. Will point to slides and freely available online material.

Useful textbooks:


*Pattern Recognition and Machine Learning*  
Christopher Bishop, Springer-Verlag 2006
Grading

- 40% for homeworks. There are 5 and you can drop 1.
- 20% for midterm [March 7]
- 20% for final [May 2nd]
- 15% for project
- 5% for class participation.
  - Piazza polls in class: bring a laptop or a phone
- Homework 0: background hwk, out today [get full credit if you turn it in]
- Homeworks 1 to 4
  - Theory/math handouts
  - Programming exercises; applying/evaluating existing learners
  - Late assignments:
    - Up to 50% credit if it’s less than 48 hrs late
    - You can drop your lowest assignment grade
- Projects: conduct a small experiment or read a couple of papers and present the main ideas or work on a small theoretical question.
  - Project presentations: April 23 and April 25
Collaboration policy (see syllabus)

• Discussion of anything is ok…
• …but the goal should be to understand better, not save work.

• So:
  – no notes of the discussion are allowed…the only thing you can take away is whatever’s in your brain.
  – you should acknowledge who you got help from/did help in your homework
Brief Overview

• **Meeting Time:** Mon, Wed, NSH 3002, 10:30 – 11:50

• **Course Staff**
  
  • Instructors:
    – Maria Florina (Nina) Balcan (ninamf@cs.cmu.edu)
  
  • TAs:
    – Kenneth Marino (kdmartino@cs.cmu.edu)
    – Colin White (crwhite@cs.cmu.edu)
    – Nupur Chatterji (nchatter@andrew.cmu.edu)
Maria-Florina Balcan: Nina

- Foundations for Modern Machine Learning
- E.g., interactive, semi-supervised, distributed, life-long learning

- Connections between learning & other fields (algorithms, algorithmic game theory)

- Program Committee Chair for ICML 2016, COLT 2014
Kenneth Marino: Kenny

- Incorporating knowledge into Computer Vision
  - Incorporating knowledge graphs
  - Learning from Wikipedia articles

- Deep Learning – non-traditional training and architectures
  - Graph Networks
  - Generative Models (VAEs and GANs)
Colin White

- 4th year PhD student advised by Nina Balcan
- Design and analysis of algorithms
- Theoretical foundations of machine learning
- Beyond worst-case analysis

Worst-case   Average case   Real-world, application-specific
Nupur Chatterji

• Senior is SCS (Undergrad)
• Minor in Machine Learning (and Economics)
• Intend to pursue ML in grad school
• Interested in the intersection between technology and healthcare
Learning Decision Trees.
Supervised Classification.

Useful Readings:
- Mitchell, Chapter 3
- Bishop, Chapter 14.4

DT learning: Method for learning discrete-valued target functions in which the function to be learned is represented by a decision tree.
**Supervised Classification: Decision Tree Learning**

**Example:** learn concept **PlayTennis** (i.e., decide whether our friend will play tennis or not in a given day)

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>Play Tennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>
Supervised Classification: Decision Tree Learning

- Each internal node: test one (discrete-valued) attribute $X_i$
- Each branch from a node: corresponds to one possible values for $X_i$
- Each leaf node: predict $Y$ (or $P(Y=1|x \in \text{leaf})$)

Example: A Decision tree for

\[ f: <\text{Outlook}, \text{Temperature}, \text{Humidity}, \text{Wind}> \rightarrow \text{PlayTennis?} \]

\[
\begin{array}{|c|c|c|c|c|}
\hline
\text{Day} & \text{Outlook} & \text{Temperature} & \text{Humidity} & \text{Wind} & \text{Play Tennis} \\
\hline
D1 & \text{Sunny} & \text{Hot} & \text{High} & \text{Weak} & \text{No} \\
D2 & \text{Sunny} & \text{Hot} & \text{High} & \text{Strong} & \text{No} \\
D3 & \text{Overcast} & \text{Hot} & \text{High} & \text{Weak} & \text{Yes} \\
D4 & \text{Rain} & \text{Mild} & \text{High} & \text{Weak} & \text{Yes} \\
D5 & \text{Rain} & \text{Cool} & \text{Normal} & \text{Weak} & \text{Yes} \\
D6 & \text{Rain} & \text{Cool} & \text{Normal} & \text{Strong} & \text{No} \\
D7 & \text{Overcast} & \text{Cool} & \text{Normal} & \text{Strong} & \text{Yes} \\
D8 & \text{Sunny} & \text{Mild} & \text{High} & \text{Weak} & \text{No} \\
D9 & \text{Sunny} & \text{Cool} & \text{Normal} & \text{Weak} & \text{Yes} \\
D10 & \text{Rain} & \text{Mild} & \text{Normal} & \text{Weak} & \text{Yes} \\
D11 & \text{Sunny} & \text{Mild} & \text{Normal} & \text{Strong} & \text{Yes} \\
D12 & \text{Overcast} & \text{Mild} & \text{High} & \text{Strong} & \text{Yes} \\
D13 & \text{Overcast} & \text{Hot} & \text{Normal} & \text{Weak} & \text{Yes} \\
D14 & \text{Rain} & \text{Mild} & \text{High} & \text{Strong} & \text{No} \\
\hline
\end{array}
\]

E.g., $x=(\text{Outlook}=\text{sunny}, \text{Temperature}=\text{Hot}, \text{Humidity}=\text{Normal}, \text{Wind}=\text{High})$, $f(x)=\text{Yes}$. 
Supervised Classification: Problem Setting

Input: Training labeled examples \( \{(x^{(i)},y^{(i)})\} \) of unknown target function \( f \)

- Examples described by their values on some set of features or attributes
  - E.g. 4 attributes: Humidity, Wind, Outlook, Temp
    - e.g., \(<\text{Humidity}=\text{High}, \text{Wind}=\text{weak}, \text{Outlook}=\text{rain}, \text{Temp}=\text{Mild}>\)
  - Set of possible instances \( X \) (a.k.a instance space)

- Unknown target function \( f : X \rightarrow Y \)
  - e.g., \( Y=\{0,1\} \) label space
  - e.g., 1 if we play tennis on this day, else 0

Output: Hypothesis \( h \in H \) that (best) approximates target function \( f \)

- Set of function hypotheses \( H=\{ h \mid h : X \rightarrow Y \} \)
  - each hypothesis \( h \) is a decision tree
Supervised Classification: Decision Trees

Suppose \( X = <x_1, \ldots, x_n> \)
where \( x_i \) are boolean-valued variables

How would you represent the following as DTs?

\[
f(x) = x_2 \ \text{AND} \ x_5?
\]

\[
f(x) = x_2 \ \text{OR} \ x_5
\]

Hwk: How would you represent \( X_2 \ X_5 \ \lor \ X_3X_4(\neg X_1) \) ?
Supervised Classification: Problem Setting

Input:  Training labeled examples \( \{(x^{(i)}, y^{(i)})\} \) of unknown target function \( f \)

- **Examples described by their values on some set of features or attributes**
  - E.g. 4 attributes: \( \text{Humidity, Wind, Outlook, Temp} \)
    - e.g., \(<\text{Humidity}=\text{High}, \text{Wind}=\text{weak}, \text{Outlook}=\text{rain}, \text{Temp}=\text{Mild}>\)
- **Set of possible instances \( X \) (a.k.a instance space)**

- **Unknown target function \( f : X \rightarrow Y \)**
  - e.g., \( Y = \{0,1\} \) label space
  - e.g., 1 if we play tennis on this day, else 0

Output:  Hypothesis \( h \in H \) that (best) approximates target function \( f \)

- **Set of function hypotheses \( H = \{ h \mid h : X \rightarrow Y \} \)**
  - each hypothesis \( h \) is a decision tree
Core Aspects in Decision Tree & Supervised Learning

How to automatically find a good hypothesis for training data?

- This is an **algorithmic** question, the main topic of computer science

When do we generalize and do well on unseen data?

- **Learning theory** quantifies ability to *generalize* as a function of the amount of training data and the hypothesis space
- **Occam’s razor**: use the *simplest* hypothesis consistent with data!

Fewer short hypotheses than long ones
- a short hypothesis that fits the data is less likely to be a statistical coincidence
- highly probable that a sufficiently complex hypothesis will fit the data
Core Aspects in Decision Tree & Supervised Learning

How to automatically find a good hypothesis for training data?

• This is an **algorithmic** question, the main topic of computer science

When do we generalize and do well on unseen data?

• **Occam’s razor**: use the *simplest* hypothesis consistent with data!

• Decision trees: if we were able to find a **small decision tree** that explains data well, then good generalization guarantees.
  
  • NP-hard [Hyafil-Rivest’76]: unlikely to have a poly time algorithm

• Very nice practical heuristics: top down algorithms, e.g, ID3
Top-Down Induction of Decision Trees

ID3: Natural greedy approach to growing a decision tree top-down (from the root to the leaves by repeatedly replacing an existing leaf with an internal node.).

Algorithm:
- Pick “best” attribute to split at the root based on training data.
- Recurse on children that are impure (e.g, have both Yes and No).