CM146
Introduction to Machine Learning
Fall 2017
Sriram Sankararaman

The instructor gratefully acknowledges Eric Eaton (UPenn), who assembled the original slides, Jessica Wu (Harvey Mudd), David Kauchak (Pomona), whose slides are also heavily used, and the many others who made their course materials freely available online.
Machine Learning is...

Machine learning is about predicting the future based on the past.

-- Hal Daume III
Machine learning is...

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 \( <P, T, E> \).

[Definition by Tom Mitchell (1998)]
Goals of the course: Learn about...

Fundamental concepts and algorithms

Common techniques/tools used
- theoretical understanding
- practical implementation
- best practices

Based on slide by David Kauchak
Administrivia
Registration

Course is currently full

Students on the waiting list will be enrolled in case some one drops.

   No guarantees though

Won’t be giving PTEs till after the first math mini-quiz.
Registration

Expect several students will drop the course.
  Course requires mathematical maturity
  Last offering did see attrition later in the quarter.
Problem set 0 + math mini-quiz intended to solve this.
  Representative of the math needed for this course.
  Course requirement.
  Graded to assess background but not part of the final grade.
  Encouraged to be honest/realistic about your background.
Prerequisites

The pillars of machine learning
    Probability and statistics
    Linear algebra
    Optimization
Prerequisites

Multivariate calculus
Linear algebra
Probability and statistics
Algorithms

Programming experience needed

Python, numpy and scikit-learn (a machine learning library for python)
Problem set 0 intended to get you up to speed
Math background review

Problem set 0 posted to self evaluate if you have the background and to help you recall concepts that you might have learned.

1. Minimum background section
2. Moderate background section

If you pass (2), you are in good shape.

If you pass (1) but not (2), you should expect to fill in the background needed (we will also cover this material in class).

If you cannot pass (1), you should fill in your math background before taking the class.
Math background review

Mini quiz (30 minutes) on Monday 10/9 intended to review your preparedness.

In-class, closed book, closed notes

Will be graded by us to give you feedback but does not count towards your final grade.

We will not grade any other problem sets/exams unless you attempt problem set 0 and math mini quiz.
Textbooks

No one textbook

Primary reference: A course in machine learning by Hal Daume III (CIML). Freely available online

Pattern recognition and machine learning by Chris Bishop (PRML)
Course format

Problem sets (aka homeworks) (50%)

Six problem sets (numbered 0 to 5)
Due at 11:59pm on the due date
Late submissions not accepted
Will be using gradescope to manage submissions (will send out submission instructions)
    All solutions must be clearly written or typed. Unreadable answers will not be graded. We encourage using LaTeX to typset answers.
    Solutions will be graded on both correctness and clarity.

You are free to discuss homework problems. However, you must write up your own solutions. You must also acknowledge all collaborators.
Course format

Mini quiz on math background (0%)
In class, closed-book and closed-notes mini quiz that will help you evaluate your background.
Does not count towards your final grade.

Exams (Mid-term: 20%, Final: 30%)
Scheduled for Nov 6 and Dec 11.
Exams are in class, closed-book and closed-notes and will cover material from the lectures and the problem sets.
No alternate or make-up exams will be administered, except for disability/medical reasons documented and communicated to the instructor prior to the exam date. In particular, exam dates and times cannot be changed to accommodate scheduling conflicts with other classes.
Course format

Final grades will be done based on a curve
Software

We will extensively use Python 2.7.x to implement ML algorithms and to run experiments. You will need to familiarize yourself with the following python packages.

- **numpy**: tools for numerical linear algebra
- **scipy**
- **scikit-learn**: tools for machine learning and data science
Forums

Piazza

Must have already got an email

Otherwise you can sign up:
  piazza.com/ucla/fall2017/cm146

Strongly encourage students to post here rather than email course staff directly (you will get a faster response this way)

If you do need to contact the staff privately, Piazza allows you to do this.
Forums

Gradescope

For managing homework and exam submissions.

For homeworks, you will need to upload pdfs of your submission to gradescope.

Code for sign up: MV88YG
Policies

Academic integrity policy

Please refer the course website
Policies

Attendance and class participation

Although not a formal component of the grade, attendance is important (and we look forward to your active participation).

If you are absent without a documented excuse, the instructor and TA will not be able to go over missed lecture material.

Video recordings: we aim to make recordings of the lectures available. You should not rely on these recordings as a substitute for lectures.
Regrade requests

Regrade requests must be made within one week after the graded homeworks have been handed out, regardless of your attendance on that day and regardless of any intervening holidays such as Memorial Day.

We reserve the right to regrade all problems for a given regrade request.
Tentative syllabus

Refer to class website

Questions?
When Do We Use Machine Learning?

ML is used when:

- Human expertise does not exist (navigating on Mars)
- Humans cannot explain their expertise (speech recognition)
- Algorithms must be customized (personalized medicine)
- Data exists to acquire expertise (genomics)

Learning is not always useful:

- There is no need to “learn” to add numbers

Slide credit: Eric Eaton
Based on slide by E. Alpaydin
A classic example of a task that requires machine learning:

It is very hard to say what makes a 2
Some more examples of tasks that are best solved by using a learning algorithm

• Recognizing patterns:
  – Facial identities or facial expressions
  – Handwritten or spoken words
  – Medical images

• Generating patterns:
  – Generating images or motion sequences

• Recognizing anomalies:
  – Unusual credit card transactions
  – Unusual patterns of sensor readings in a nuclear power plant

• Prediction:
  – Future stock prices or currency exchange rates

Slide credit: Geoffrey Hinton
Sample Applications

- Web search
- Computational biology
- Finance
- E-commerce
- Space exploration
- Robotics
- Information extraction
- Social networks
- Debugging software
- [Your favorite area]

Slide credit: Pedro Domingos
Defining the Learning Task

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

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

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

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

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

Slide credit: Ray Mooney
State of the Art Applications of Machine Learning
Autonomous Cars

- Nevada made it legal for autonomous cars to drive on roads in June 2011
- As of 2013, four states (Nevada, Florida, California, and Michigan) have legalized autonomous cars

Slide credit: Eric Eaton
Autonomous Car Technology

Laser Terrain Mapping

Learning from Human Drivers

Adaptive Vision

Slide credit: Eric Eaton

[Source: Sebastian Thrun’s multimedia website]
Deep Learning in the Headlines

Is Google Cornering the Market on Deep Learning?
A cutting-edge corner of science is being wooed by Silicon Valley, to the dismay of some academics.
by Antonio Regalado on January 29, 2014

How much are a dozen deep-learning researchers worth? Apparently, more than $400 million.
This week, Google reportedly paid that much to acquire DeepMind Technologies, a startup based in Cambridge, England.

Deep Learning’s Role in the Age of Robots
BY JULIAN GREEN, JETPAC 05.02.14 2:56 PM

BloombergBusinessweek
Technology
Acquisitions
The Race to Buy the Human Brains Behind Deep Learning Machines
by Ashlee Vance  January 27, 2014

intelligence projects. “DeepMind is bona fide in terms of its research capabilities and depth,” says Peter Lee, who heads Microsoft Research.

According to Lee, Microsoft, Facebook (FB), and Google find themselves in a battle for deep learning talent. Microsoft has gone from four full-time deep learning experts to 70 in the past three years. “We would have more if the talent was there to

Slide credit: Eric Eaton
Scene Labeling via Deep Learning

Slide credit: Eric Eaton
[Source: Farabet et al. ICML 2012, PAMI 2013]
Impact of Deep Learning in Speech Technology

Slide credit: Li Deng, MS Research
Types of Learning
Types of Learning

• **Supervised (inductive) learning**: Learn with a teacher
  – Given: labeled training instances (or examples)
  – Goal: learn mapping that predicts label for test instance

• **Unsupervised learning**: Learn without a teacher
  – Given: unlabeled inputs
  – Goal: learn some intrinsic structure in inputs

• **Reinforcement learning**: Learn by interacting
  – Given agent interacting in environment (having set of states)
  – Learn policy (state to action mapping) that maximizes agent’s reward
Supervised Learning: Regression

- Given \((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\)
- Learn a function \(f(x)\) to predict \(y\) given \(x\)
  - \(y\) is real-valued == regression

Slide credit: Eric Eaton
Supervised Learning: Classification

- Given \((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\)
- Learn a function \(f(x)\) to predict \(y\) given \(x\)
  - \(y\) is categorical == classification

Based on slide by Eric Eaton
[example originally by Andrew Ng]
Supervised Learning

- $x$ can be multi-dimensional
  - each dimension corresponds to an attribute

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

Based on slide by Eric Eaton
[example originally by Andrew Ng]
Unsupervised Learning

• Given $x_1, x_2, \ldots, x_n$ (without labels)
• Output hidden structure behind the $x$’s
  – e.g., clustering
Unsupervised Learning

Genomics application: group individuals by genetic similarity
Unsupervised Learning

Independent component analysis – separate a combined signal into its original sources

Slide credit: Eric Eaton
Audio from http://www.ism.ac.jp/~shiro/research/blindsep.html
Image credit: statsoft.com
Reinforcement Learning

• Given sequence of states and actions with (delayed) rewards
• Learn policy that maximizes agent’s reward
• Examples:
  – Game playing
  – Robot in maze
Reinforcement Learning

Backgammon

Given sequences of moves and whether or not the player won at the end, learn to make good moves

Slide credit: David Kauchak
Framing a Learning Problem
Representing instances/examples

What is an instance?
How is it represented?

instances

features

feat_1, feat_2, feat_3, feat_4, ...
red, round, leaf, 3oz, ...

feat_1, feat_2, feat_3, feat_4, ...
green, round, no leaf, 4oz, ...

feat_1, feat_2, feat_3, feat_4, ...
yellow, curved, no leaf, 4oz, ...

feat_1, feat_2, feat_3, feat_4, ...
green, curved, no leaf, 5oz, ...

How our algorithms actually “view” the data

Features are the questions we can ask about the instances
During learning/training/induction, learn a model of what distinguishes apples and bananas based on the features.

The classifier classifies a new instance based on the features.
Learning algorithm

• Learning is about **generalizing** from training data

• What does this **assume** about training and test set?

![Training data](image1)
![Test set](image2)
![Training data](image3)

Not always the case, but we’ll often assume it is!

Based on slide by David Kauchak
• We care about the performance of the learning algorithm on test data (generalization ability).
• How we measure performance depends on the problem we are trying to solve
• The training and test data should be strongly related.
More Technically...

• We start with a loss function \( L(y, \hat{y}) \)

• Tells us how bad the system’s prediction of \( y \) is compared to the true value of \( y \)
  – A loss function for regression (squared loss)
    \[
    L(y, \hat{y}) = (y - \hat{y})^2
    \]
  – A loss function for classification
    \[
    L(y, \hat{y}) = \begin{cases} 
      0 & \text{if } y = \hat{y} \\
      1 & \text{otherwise}
    \end{cases}
    \]
More Technically...

• We are going to use the *probabilistic model* of learning

• There is some *unknown* probability distribution $p$ over instance/label pairs called the *data generating distribution*
Learning problem

Defined by

- Loss function: measures performance
- Data generating distribution: what data do we expect to see (characterizes experience)
Learning problem

Problem Setting

• Set of possible instances $X$
• Set of possible labels $Y$
• Unknown target function $f : X \rightarrow Y$
• Set of function hypotheses $H = \{ h \mid h : X \rightarrow Y \}$

Input: Training instances drawn from data generating distribution $p$

\[ \{(x_i, y_i)\}_{i=1}^n = \{(x_1, y_1), \ldots, (x_n, y_n)\} \]

Output: Hypothesis $h$ in $H$ that best approximates $f$
Learning problem

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

\( h \) should do well (as measured by the loss) on future instances

Formally, \( h \) should have **low expected (test) loss/Risk**

\[
\mathbb{E}_{(x,y) \sim p} [L(y, h(x))] = \sum_{x,y} p(x,y)L(y, h(x))
\]

Problem?

We don’t know what \( p \) is

But we are given samples drawn from \( p \)
Learning problem

We instead approximate the risk by the training error/empirical risk

$$\frac{1}{n} \sum_{i=1}^{n} L(y_i, h(x_i))$$

When is this reasonable?

**Both** the training data **and** the test set are generated based on this distribution

Problem?

Can make the training error zero by **memorizing**
Example Regression Problem

- Consider simple regression dataset
  - \( f : X \rightarrow Y \)
  - \( x \in \mathbb{R} \)
  - \( y \in \mathbb{R} \)

- **Question 1:** How should we pick the hypothesis space \( H \)?
- **Question 2:** How do we find the best \( h \) in this space?
Hypothesis Space: Degree-M Polynomials

- Infinitely many hypotheses
- Which one is best?

Based on slide by David Sontag
Images from Bishop [PRML]
• For regression, common choice is squared loss

$$L(y_i, h(x_i)) = (y_i - h(x_i))^2$$

• *Empirical loss* of function $h$ applied to training data is then

$$\frac{1}{n} \sum_{i=1}^{n} L(y_i, h(x_i)) = \frac{1}{n} \sum_{i=1}^{n} (y_i - h(x_i))^2$$

Based on slide by David Sontag
Images from Bishop [PRML]
Learning problem

The fundamental difficulty of machine learning

We have access to the training error but really care about the expected loss

Our learned function needs to generalize beyond the training data
Key Issues in Machine Learning

**Representation** : How do we choose a hypothesis space?
- Often we use *prior knowledge* to guide this choice
- The ability to answer the next two questions also affects choice

**Optimization** : How do we find the best hypothesis within this space?
- This is an *algorithmic* question, at the intersection of computer science and optimization research.

**Evaluation** : How can we gauge the accuracy of a hypothesis on unseen testing data?
- The previous example showed that choosing the hypothesis which simply minimizes training set error (i.e. empirical loss) **is not optimal**
- This question is the main topic of *learning theory*

Based on slides by Eric Eaton and by David Sontag
More Issues in Machine Learning

**Representation**: How do we choose a hypothesis space?
- Which spaces have been useful in practical applications and why?

**Optimization**: How do we find the best hypothesis within this space? (the **algorithmic** question)
- Are some learning problems computationally intractable? (the **computational** question)

**Evaluation**: How do we find the best hypothesis within this space?
- How can we have confidence in the results? (the **statistical** question)

**Formulation**: How can we formulate application problems as machine learning problems? (the **engineering** question)

Based on slides by Pedro Domingos
ML in Practice

- Understand domain, prior knowledge, and goals
- Data integration, selection, cleaning, pre-processing.
- Learn models
- Interpret results

Loop

Slide credit: Eric Eaton
Based on a slide by Pedro Domingos
What we will Cover in this Course

• **Supervised learning**
  – Decision tree
  – Perceptron
  – Linear regression
  – Logistic regression
  – Support vector machines & kernel methods
  – Ensemble methods
  – PCA
  – Clustering
  – Hidden Markov Models

• **Experimental evaluation**
  – Cross-validation
  – Metrics
  – Real datasets!
Summary

Next class: Review of probability and statistics

Problem Set 0 will be released today