UVA CS 6316 / CS 4501
– Fall 2015 :
Machine Learning

Lecture 1: Introduction

Dr. Yanjun Qi

University of Virginia
Department of
Computer Science
Welcome

• CS 6316/4501 Machine Learning
  – MoWe 3:30pm-4:45pm,  
  – Olsson Hall 120

• http://www.cs.virginia.edu/yanjun/teach/2016f

• Your UVA collab: Course 6316-4501 page
Today

- Course Logistics
- My background
- Basics and rough content plan
- Application and History
Course Staff

- Instructor: Prof. Yanjun Qi
  - QI: /ch ee/
  - You can call me “professor”, “professor Jane”, “professor Qi”;

- TA office hours: Mon & Wed 5:30pm-6:30pm @ Rice 504
- My office hours: Mon 5pm-6pm @ Rice 503
Course Logistics

• Course email list has been setup. You should have received emails already!

• Policy, the grade will be calculated as follows:
  - Assignments (55%, Six total, each ~9%)
  - Quizzes / Exam Sample Practices (5%)
  - Midterm exam (20%)
  - Final exam (20%)
Course Logistics

• Midterm: Oct, 75mins in class
• Final: Dec, 75mins in class

• Six assignments (each 9% to 10%)
  — Three extension days policy (check course website)

• In-class quizzes / Exam sample practice (total 5%)

9/1/16
Course Logistics

• Policy,
  – Homework should be submitted electronically through UVaCollab
  – Homework should be finished individually
  – Due at midnight on the due date
  – In order to pass the course, the average of your midterm and final must also be "pass".
Late Homework Policy

• Each student has **three** extension days to be used at his or her own discretion throughout the entire course. Your grades would be discounted by 15% per day when you use these 3 late days. You could use the 3 days in whatever combination you like. For example, all 3 days on 1 assignment (for a maximum grade of 55%) or 1 each day over 3 assignments (for a maximum grade of 85% on each). After you've used all 3 days, you cannot get credit for anything turned in late.
Course Logistics

• Text books for this class is:
  – NONE

• My slides – if it is not mentioned in my slides, it is not an official topic of the course
Course Logistics

- **Background Needed**
  - Calculus, Basic linear algebra, Basic probability and Basic Algorithm
  - Statistics is recommended.
  - Students should already have good programming skills, i.e. **python** is required for all programming assignments
  - We will review “linear algebra” and “probability” in class
Today

- Course Logistics
- My background
- Basics and rough content plan
- Application and History
About Me

• Education:
  – PhD from School of Computer Science, Carnegie Mellon University (@ Pittsburgh, PA) in 2008
  – BS from Department of Computer Science, Tsinghua Univ. (@ Beijing, China)
    • My accent PATTERN : /l/, /n/, /ou/, /m/

• Research interests:
  – Machine Learning, Biomedical applications
About Me

• Five Years’ of Industry Research Lab in the past :
  – 2008 summer – 2013 summer, Research Scientist in IT industry (Machine Learning Department, NEC Labs America @ Princeton, NJ)
  – 2013 Fall – Present, Assistant Professor, Computer Science, UVA
Today

- Course Logistics
- My background
- Basics and Rough content plan
- Application and History
OUR DATA-RICH WORLD

• Biomedicine
  – Patient records, brain imaging, MRI & CT scans, …
  – Genomic sequences, bio-structure, drug effect info, …

• Science
  – Historical documents, scanned books, databases from astronomy, environmental data, climate records, …

• Social media
  – Social interactions data, twitter, facebook records, online reviews, …

• Business
  – Stock market transactions, corporate sales, airline traffic, …

• Entertainment
  – Internet images, Hollywood movies, music audio files, …
BIG DATA CHALLENGES

- Data capturing (sensor, smart devices, medical instruments, et al.)
- Data transmission
- Data storage
- Data management
- High performance data processing
- Data visualization
- Data security & privacy (e.g. multiple individuals)
- .......

- Data analytics
  - How can we analyze this big data wealth?
  - E.g. Machine learning and data mining
BASICS OF MACHINE LEARNING

• “The goal of machine learning is to build computer systems that can learn and adapt from their experience.” – Tom Dietterich

• “Experience” in the form of available data examples (also called as instances, samples)

• Available examples are described with properties (data points in feature space X)
e.g. **SUPERVISED LEARNING**

- Find function to map **input** space $X$ to **output** space $Y$ 
  
  \[ f : X \rightarrow Y \]

- So that the **difference** between $y$ and $f(x)$ of each example $x$ is small.

**e.g.**

<table>
<thead>
<tr>
<th>Input $X$</th>
<th>I believe that this book is not at all helpful since it does not explain thoroughly the material. It just provides the reader with tables and calculations that sometimes are not easily understood...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output $Y$</td>
<td>-1</td>
</tr>
</tbody>
</table>

Output $Y$: \{1 / Yes, -1 / No\}

**e.g.** Is this a positive product review?
e.g. SUPERVISED Linear Binary Classifier

\[ f(x, w, b) = \text{sign}(w \cdot x + b) \]

- \( w \cdot x + b > 0 \) denotes +1 point
- \( w \cdot x + b < 0 \) denotes -1 point
- \( w \cdot x + b = 0 \) denotes future points

"Predict Class = +1" zone
"Predict Class = -1" zone

Courtesy slide from Prof. Andrew Moore’s tutorial
Basic Concepts

- **Training** (i.e. learning parameters $\{w, b\}$)
  - Training set includes
    - available examples $x_1, \ldots, x_L$
    - available corresponding labels $y_1, \ldots, y_L$

  - Find $(w, b)$ by minimizing loss (i.e. difference between $y$ and $f(x)$ on available examples in training set)

$$ (W, b) = \arg\min_{w, b} \sum_{i=1}^{L} \ell(f(x_i), y_i) $$
Basic Concepts

- **Testing** (i.e. evaluating performance on “future” points)
  - Difference between true $y_i$ and the predicted $f(x_i)$ on a set of testing examples (i.e. testing set)
  - Key: example $x_i$ not in the training set

- **Generalisation**: learn function / hypothesis from past data in order to “explain”, “predict”, “model” or “control” new data examples
Basic Concepts

• Loss function
  – e.g. hinge loss for binary classification task
    \[ \sum_{i=1}^{L} \ell(f(x_i), y_i) = \sum_{i=1}^{L} \max(0, 1 - y_i f(x_i)) \]
  – e.g. pairwise ranking loss for ranking task (i.e. ordering examples by preference)

• Regularization
  – E.g. additional information added on loss function to control model
    \[ C \sum_{i=1}^{L} \ell(f(x_i), y_i) + \frac{1}{2} \| w \|^2 \]
TYPICAL MACHINE LEARNING SYSTEM

- Low-level sensing
- Pre-processing
- Feature Extract
- Feature Select
- Inference, Prediction, Recognition
- Label Collection
- Evaluation

\[ f : X \rightarrow Y \]
“Big Data” Challenges for Machine Learning

- Large size of samples
- High dimensional features

Not the focus, will be covered in advanced-level course
Large-Scale Machine Learning: SIZE MATTERS

- One thousand data instances
- One million data instances
- One billion data instances
- One trillion data instances

Those are not different numbers, those are different mindsets !!!
BIG DATA CHALLENGES FOR MACHINE LEARNING

Most of this course

The variations of both $X$ (feature, representation) and $Y$ (labels) are complex!

✓ Complexity of $X$
✓ Complexity of $Y$
TYPICAL MACHINE LEARNING SYSTEM

Low-level sensing → Pre-processing → Feature Extract → Feature Select → Inference, Prediction, Recognition

Data Complexity of X

Data Complexity of Y

Label Collection → Evaluation

\( f : X \rightarrow Y \)
UNSUPERVISED LEARNING:
[ COMPLEXITY OF Y ]

• No labels are provided (e.g. No Y provided)
• Find patterns from unlabeled data, e.g. clustering

E.g. clustering => to find “natural” grouping of instances given un-labeled data
Many prediction tasks involve output labels having structured correlations or constraints among instances.

<table>
<thead>
<tr>
<th>Structured Dependency between Examples</th>
<th>Sequence</th>
<th>Tree</th>
<th>Grid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input $X$</td>
<td>APAFSVSPASGACGPECA...</td>
<td>The dog chased the cat</td>
<td></td>
</tr>
<tr>
<td>Output $Y$</td>
<td>CEEEEBCCCCCHHHCCC...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Many more possible structures between $y_i$, e.g. spatial, temporal, relational...
STRUCTURAL INPUT : Kernel Methods

[ COMPLEXITY OF X ]

Vector vs. Relational data

e.g. Graphs, Sequences, 3D structures,

Original Space  Feature Space
MORE RECENT: FEATURE LEARNING

[ COMPLEXITY OF X ]

Deep Learning

Supervised Embedding

Layer-wise Pretraining
DEEP LEARNING / FEATURE LEARNING: [COMPLEXITY OF X]

**Feature Engineering**
- Most critical for accuracy
- Account for *most of the computation* for testing
- Most time-consuming in development cycle
- Often *hand-craft* and *task dependent* in practice

**Feature Learning**
- Easily *adaptable to new* similar tasks
- Layerwise representation
- Layer-by-layer unsupervised training
- Layer-by-layer supervised training
**Deep Learning**

With massive amounts of computational power, machines can now recognize objects and translate speech in real time. Artificial intelligence is finally getting smart.

---

**Temporary Social Media**

Messages that quickly self-destruct could enhance the privacy of online communications and make people freer to be spontaneous.

---

**Prenatal DNA Sequencing**

Reading the DNA of fetuses will be the next frontier of the genomic revolution. But do you really want to know about the genetic problems or musical aptitude of your unborn child?

---

**Additive Manufacturing**

Skeptical about 3-D printing? GE, the world’s largest manufacturer, is on the verge of using the technology to make jet parts.

---

**Baxter: The Blue-Collar Robot**

Rodney Brooks’s newest creation is easy to interact with, but the complex innovations behind the robot show just how hard it is to get along with people.

---

**Memory Implants**

A maverick neuroscientist believes he has deciphered the code by which the brain forms long-term memories. Next: testing a prosthetic implant for people suffering from long-term memory loss.

---

**Smart Watches**

The designers of the Pebble watch realized that a mobile phone is more useful if you don’t have to take it out of your pocket.

---

**Ultra-Efficient Solar Power**

Doubling the efficiency of a solar cell would completely change the economics of renewable energy. Nanotechnology just might make it possible.

---

**Big Data from Cheap Phones**

Collecting and analyzing information from simple cell phones can provide surprising insights into how people move about and behave – and even help us understand the spread of diseases.

---

**Supergrids**

A new high-power circuit breaker could finally make highly efficient DC power grids practical.
Course Content Plan ➔

Five major sections of this course

- Regression (supervised)
- Classification (supervised)
- Unsupervised models
- Learning theory
- Graphical models
scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which set of categories a new observation belong to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, ...

Examples

Regression

Predicting a continuous value for a new example.

Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, ...

Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, ...

Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization

Examples

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics

Examples

Preprocessing

Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction

Examples

Scikit-learn algorithm cheat-sheet

classification

kernel approximation

SVC
Ensemble Classifiers
RNeighborhood Classifier
SGD Classifier

Naive Bayes
Text Data
Linear SVC

<100K samples

Do you have labeled data?

number of categories known

<10K samples

MiniBatch KMeans
MeanShift
VBGMM

<10K samples

Spectral Clustering
GMM

<10K samples

tough luck

predicting structure

predicting a category

predicting a quantity

get more data

>50 samples

<100K samples

regression

SGD Regressor

Lasso
ElasticNet

SVR(kernel="rbf")

Ensemble Regressors

RidgeRegression
SVR(kernel="linear")

few features should be important

<100K samples

<10K samples

Randomized PCA

Isomap
Spectral Embedding

LLE

predicting a quantity

<10K samples

<10K samples

dimensionality reduction

back

scikit-learn
Scikit-learn: Regression

Linear model fitted by minimizing a regularized empirical loss with SGD

Regression

SGD Regressor

Lasso
ElasticNet

SVR(kernel='rbf')
EnsembleRegression

RidgeRegression
SVR(kernel='linear')

<100K samples

few features should be important

Linear Regression + Variations

NOT WORKING

YES

NO

YES

NO

Dr. Yanjun Qi / UVA CS 6316-4501 / f16
Scikit-learn: Classification

To combine the predictions of several base estimators built with a given learning algorithm in order to improve generalizability/robustness over a single estimator. (1) averaging/bagging (2) boosting

Approximate the explicit feature mappings that correspond to certain kernels

Linear classifiers (SVM, logistic regression...) with SGD training.
Unsupervised Models

- **Kmeans + GMM**
- **Spectral Clustering + GMM**
- **KMeans**
- **<10K samples**
- **<10K samples**
- **MiniBatch KMeans**
- **MeanShift**
- **VBGMM**
- **<10K samples**
- **number of categories known**
- **<10K samples**

- **Bayes-Net HMM**
- **Basic PCA**
- **Randomized PCA**
- **Isomap**
- **Spectral Embedding**
- **LLE**
- **kernel approximation**

- **dimensionality reduction**
- **tough luck**
- **predicting structure**
- **just looking**
- **<10K samples**
- **<10K samples**
- **NOT WORKING**
- **YES**
- **NO**

Dr. Yanjun Qi / UVA CS 6316-4501 / f16
Summary

• This is not a course about learning to use toolbox

• We focus on learning principles, mathematical formulation, algorithm design and learning theory.
Today

- Course Logistics
- My background
- Basics and rough content plan
- Application and History
What can we do with the data wealth?

-REAL-WORLD IMPACT-

- Business efficiencies
- Scientific breakthroughs
- Improve quality-of-life:
  - healthcare,
  - energy saving / generation,
  - environmental disasters,
  - nursing home,
  - transportation,
  - ...

Dr. Yanjun Qi / UVA CS 6316-4501 / f16

9/1/16
When to use **Machine Learning (Adapt to / learn from data)**?

1. **Extract knowledge from data**
   - Relationships and correlations can be hidden within large amounts of data
   - The amount of knowledge available about certain tasks is simply too large for explicit encoding (e.g. rules) by humans

2. **Learn tasks that are difficult to formalise**
   - Hard to be defined well, except by examples, e.g., face recognition

3. **Create software that improves over time**
   - New knowledge is constantly being discovered.
   - Rule or human encoding-based system is difficult to continuously re-design “by hand”.
MACHINE LEARNING IS CHANGING THE WORLD

Data:
- Past pregnancy: no
- Fetal frequency: no
- Mother: 40
- Ectopic pregnancy: no
- Quadruplet/Quintuplet birth: no
- Ultrasound 1: Elective C-section
- Emergency C-section

One of 18 learned rules:
If No previous vaginal delivery, and Abnormal 2nd Trimester Ultrasound, and Malpresentation at admission
Then Probability of Emergency C-Section is 0.5

Over training data: 26/41 = 0.63,
Over test data: 12/20 = 0.60

Mining Databases

Speech Recognition

Text analysis

Object recognition

Many more!
MACHINE LEARNING IN COMPUTER SCIENCE

• Machine learning is already the preferred approach for
  – Speech recognition, natural language processing
  – Computer vision
  – Medical outcome analysis
  – Robot control …

• Why growing?
  – Improved machine learning algorithm
  – Increased data capture, new sensors, networking
  – Systems/Software too complex to control manually
  – Demand to self-customization for user, environment, ….
RELATED DISCIPLINES

• Artificial Intelligence
• Data Mining
• Probability and Statistics
• Information theory
• Numerical optimization
• Computational complexity theory
• Control theory (adaptive)
• Psychology (developmental, cognitive)
• Neurobiology
• Linguistics
• Philosophy
What are the goals of AI research?

- Artifacts that THINK like HUMANS
- Artifacts that THINK RATIONALLY
- Artifacts that ACT like HUMANS
- Artifacts that ACT RATIONALLY
How can we build more intelligent computer / machine?

- Able to
  - perceive the world
  - understand the world

- This needs
  - Basic speech capabilities
  - Basic vision capabilities
  - Language/semantic understanding
  - User behavior / emotion understanding
  - Able to think??
How can we build more intelligent computer / machine?

R2-D2 and C-3PO

@ Star Wars – 1977

to serve human beings, and fluent in "over six million forms of communication"
How can we build more intelligent computer / machine?

Jeopardy Game
➡ Requires a Broad Knowledge Base

IBM Watson

given an artificial intelligence computer system capable of answering questions posed in natural language developed in IBM's DeepQA project.
How can we build more intelligent computer / machine?

- Apple Siri / Amazon Echo
  - an intelligent personal assistant and knowledge navigator
How can we build more intelligent computer / machine? : Objective Recognition / Image Labeling

**ImageNet**: an image database organized according to the **WordNet**

**LSVRC**: Large Scale Visual Recognition Challenge based on ImageNet.

[ training on 1.2 million images [X] vs. 1000 different word labels [Y] ]

Deep Convolution Neural Network (CNN) won (as Best systems) on “very large-scale” ImageNet competition 2012 / 2013 / 2014
How can we build more intelligent computer / machine? : Objective Recognition / Image Labeling

- 2013, Google Acquired Deep Neural Networks Company headed by Utoronto “Deep Learning” Professor Hinton
- 2013, Facebook Built New Artificial Intelligence Lab headed by NYU “Deep Learning” Professor LeCun
- 2016, Google's DeepMind defeats legendary Go player Lee Se-dol in historic victory
Detour: planned programming assignments

• HW: Semantic language understanding (sentiment classification on movie review text)

• HW: Visual object recognition (labeling images about handwritten digits)

• HW: Audio speech recognition (unsupervised learning based speech recognition task)
Today Recap

- Course Logistics
- My background
- Basics and rough content plan
- Application and History
Next lesson: Machine Learning in a Nutshell

ML grew out of work in AI

Optimize a performance criterion using example data or past experience,

Aiming to generalize to unseen data

Next lesson: Review of linear algebra and basic calculus
References

- Prof. Andrew Moore’s tutorials
- Prof. Raymond J. Mooney’s slides
- Prof. Alexander Gray’s slides
- Prof. Eric Xing’s slides
- http://scikit-learn.org/
- Prof. M.A. Papalaskar’s slides