Machine Learning

Section (“Canale”) 2: x5-x9

Fabio Vandin

September 25th, 2017
Machine Learning

Section (“Canale”) 2: IDs ending with x5-x9

IMPORTANT: if you do not have a Laurea Magistrale, consider your Laurea Triennale ID (you do not need to consider your Laurea Magistrale ID later on!)

6 credits:

• 48 hours in class lectures
• 46 hours: lectures
• 2 hours: lab (mandatory – intro to Python)
• 6 hours: homeworks assistance (not mandatory)
• **102 hours individual study**

Everything (lectures, exams, homeworks) in English!
Course website:

- da elearning.dei.unipd.it:
  MACHINE LEARNING - APPRENDIMENTO AUTOMATICO Numerosita` canale 2 (5-9) 17/18

Register today: registration possible only until October 13th!
Lectures: When and where

Monday 10:15-12:15, room Ce.
Options:
   1. 10:30-11:15, 15 mins break, 11:30-12:15
   2. 10:30-11:15, 10 mins break, 11:25-12:10
   3. 10:20-11:05, 15 mins break, 11:20-12:05

Tuesday 10:15-12:15, room Ce
Options:
   1. 10:30-11:15, 15 mins break, 11:30-12:15
   2. 10:30-11:15, 10 mins break, 11:25-12:10
   3. 10:20-11:05, 15 mins break, 11:20-12:05
Labs and Homeworks

Not compulsory but HIGHLY RECOMMENDED

1. **Intro to Python:** Monday October 23th (Room Te-Ue) [compulsory]
2. **Linear models for regression and classification**
   - Tuesday October 31th: first homework will be released
   - Monday November 6th: we skip lecture, lab assistance for those interested (Room Te-Ue)
   - Tuesday November 14th: first homework due.
3. **Regularization**
   - Tuesday November 28th: second homework will be released
   - Monday December 4th: we skip lecture, lab assistance for those interested (Room Te-Ue)
   - Tuesday December 12th: second homework due.
4. **Non-linear models and unsupervised learning**
   - Tuesday December 19th: third homework will be released
   - Monday January 15th: we skip lecture, lab assistance for those interested (Room Te-Ue)
   - Tuesday January 23rd: third homework due.
Grading

Lab experience [not compulsory]: three homeworks. Up to 3 points as a bonus on the written test grade.

Written test: see sample tests from last year in elearning
  • will be graded on a scale from 0 to 30L.

Final Grade = grades written test + lab experiences

Example
2.66 lab + 24.5 written test = 27 final grade
Final Exam: dates

1. Monday, January 29th, 2018
   • time: 3.00pm, rooms: Ke,Ve
   • grading discussion and oral exams (if needed): Friday, February 2nd, 9:30am, room Ce

2. Monday, February 12th, 2018
   • time: 3.00pm, rooms: Ke,Ve
   • grading discussion and oral exams (if needed): Monday, February 19th, 9:30am, room Ce

3. Tuesday, June 26th, 2018
   • time: 3.00pm, rooms: Ke,Ve
   • grading discussion and oral exams (if needed): Monday, July 2nd, 9:30am, room Ce

4. Monday, September 17th, 2018
   • time: 3.00pm, rooms: Ke,Ve
   • grading discussion and oral exams (if needed): Friday, September 21st, 9:30am, room Ce

Duration: probably 2.5 hours
Material

Main Book

Will relate material in class to the book as much as possible

PDF available from the authors
Material (2)

Other Books (NOT Mandatory)

  
  **PDF from authors online**
  


Part of other books may be used: will provide handouts whenever possible...

Additional material: course website (*elearning.dei.unipd.it*)

- slides: published after lectures
- links, etc.

If you are missing background notions (probability, algebra): ask me for material!!!
Programming and more

Language:  

Some libraries: scikit-learn, numpy,...
  •  Will post info on the course website: keep checking!

Useful tool:
  •  Jupyter notebook:
    •  http://jupyter.readthedocs.io/en/latest/install.html
    •  Allows for a mix of text and code;
    •  If you install it through Anaconda as suggested on Jupyter website, you get a lot of other packages/libraries for ML, visualization, etc.
    •  https://www.anaconda.com/download/
Why Python

Easy! (…)

A lot of support for ML!

Used in industry, academia, research labs…
Homework 0

1. Go home and install Anaconda (version with Python 2.7):
   • https://www.continuum.io/downloads

2. Go through the following tutorial:
   – http://cs231n.github.io/python-numpy-tutorial/

3. Get used to Jupyter notebooks

4. Go through the tutorial in Jupyter notebooks and in script mode:
   • http://cs231n.github.io/python-numpy-tutorial/
New Rules for Accepting/Rejecting Grades

After grades are published in Uniweb, the students have 7 days (automatically computed by the system) to reject the grade.

Within these 7 days the student can change the status (reject/don’t reject) multiple times.

If after 7 days the grade is not rejected, it is automatically accepted.

The lecturer can register the grades only after 7 days. If there urgent requests, these need to go through administrative offices.
Lecturer

Fabio Vandin, Associate Professor, DEI (Department of Information Engineering)

Email: fabio.vandin@unipd.it

Website: www.dei.unipd.it/~vandinfa

Office: 410 (4th floor, DEI/G), phone: 049-827-7946

Office hours: Wednesday, 16:30-18:30, office 410, by request (email or after class)
Lecturer

CV

• Laurea Triennale in Computer Engineering (2004)
• Laurea Specialistica in Computer Engineering (2006)
• Ph.D. at DEI (2010)
• 2010-May/2015: Researcher/Assistant Prof.
  • Brown University (USA)
  • University of Southern Denmark

Research Interests:

• Methods: algorithms for machine learning, data mining, and big data
• Applications: Biology, Medicine, Social Networks, ...
Example: Spam Filtering

Question: Is this e-mail useful (ham) or spam?

Challenge: There is no simple universal rule to define spam.

(Noisy) data: messages previously marked as spam.

Challenge: Spammers evolve to counter filter innovations.
Example: Movie Rating Prediction

Prediction: What will be the user rating on a movie?

Challenge: Must target current user preferences

Noisy training data: ratings of all users on movies

Challenge: Data is very sparse = each user rates very few movies!
Netflix Challenge

Given: data set of 100,480,507 ratings that 480,189 users gave to 17,770 movies.

Goal: beat Netflix prediction by $\geq 10\%$.

Prize: $1,000,000$

Oct. 2006: challenged open by Netflix

Sept. 2009: prize awarded!
Example: Personalized Therapy

Given: molecular activity of genes in a patient

Prediction: how will a patient with given gene activity respond to a therapy?

Data = matrix of real numbers
- rows = genes
- columns = patient
- value = activity of gene in a patient
- response of patient to a therapy (real number)

Challenge:
- noisy data
- more genes than patients
Computer Vision
Object Recognition
Fun & curiosity

DeepMind: AI company *(developed software which “learn” to play videogames)* acquired by Google in 2014 (now Google DeepMind) for 500 M$ !!!

AlphaGo, developed by GoogleDeepMind in October 2015 *beats* European Champion of Go
Google masters Go

Deep-learning software excels at complex ancient board game.

BY ELIZABETH GIBNEY

A computer has beaten a human professional for the first time at Go — an ancient board game that has long been viewed as one of the greatest challenges for artificial intelligence (AI).

The best human players of chess, draughts and backgammon have all been outplayed by computers. But a hefty handicap was needed for computers to win at Go. Now Google’s London-based AI company, DeepMind, claims that its machine has mastered the game.

DeepMind’s program AlphaGo beat Fan Hui, the European Go champion, five times out of five in tournament conditions, the firm reveals in research published in *Nature* on 27 January. It also defeated its silicon-based rivals, winning 99.8% of games against the current best programs. The program has yet to play the Go equivalent of a world champion, but a match against South Korean professional Lee Sedol, considered by many to be the world’s strongest player, is scheduled for March. “We’re pretty confident,” says DeepMind co-founder Demis Hassabis.

“This is a really big result,” it’s huge,” says Rémi Coulom, a programmer in Lille, France, who designed a commercial Go program called Crazy Stone. He had thought computer mastery of the game was a decade away.

The IBM chess computer Deep Blue, which famously beat grandmaster Garry Kasparov in 1997, was explicitly programmed to win at the game. But AlphaGo was not preprogrammed to play Go; rather, it learned using a general-purpose algorithm that allowed it to interpret the game’s patterns, in a similar way to how a DeepMind program learned to play 49 different arcade games.

This means that similar techniques could be applied to other AI domains that require recognition of complex patterns, long-term planning and decision-making, says Hassabis. “A lot of the things we’re trying to do in the world come under that rubric,” Examples are using medical images to make diagnoses or treatment plans, and improving climate-change models.
Why Study ML?

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

“Machine learning is the next Internet” (Tony Tether, Former Director, DARPA)

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

“Web rankings today are mostly a matter of machine learning” (Prabhakar Raghavan, Vice President of Engineering at Google)

“Machine learning is going to result in a real revolution” (Greg Papadopoulos, Former CTO, Sun)
Questions?
Prediction: What will be the user rating on a movie?

Challenge: Must target current user preferences

Noisy training data: ratings of all users on movies

Challenge: Data is very sparse = each user rates very few movies!
Example: Simplified

Movie has 2 features:

• fraction of the movie with action
• fraction of the movie with romance

Rating is only LIKED or NOT LIKED

Prediction: will the user like a movie?

The essence of learning:

• A pattern exists
• We do not know it, i.e. we cannot pin it down mathematically or with very simple rules
• We have data to try to “learn” it
A Formal Model (Statistical Learning)

We have a learner (us, or the machine) has access to:

1. **Domain set** $\mathcal{X}$: set of all possible objects to make predictions about
   - domain point $x \in \mathcal{X} = \text{instance}$, usually represented by a vector of features
   - $\mathcal{X}$ is the instance space

2. **Label set** $\mathcal{Y}$: set of possible labels.
   - often two labels, e.g. $\{-1, +1\}$ or $\{0, 1\}$

3. **Training data** $S = ((x_1, y_1), \ldots, (x_m, y_m))$: finite sequence of labeled domain points, i.e. pairs in $\mathcal{X} \times \mathcal{Y}$
   - this is the learner’s input
   - $S$: training example or training set
A Formal Model

4 Learner's output $h$: prediction rule $h : \mathcal{X} \to \mathcal{Y}$
   - also called predictor, hypothesis, or classifier
   - $A(S)$: prediction rule produced by learning algorithm $A$ when training set $S$ is given to it
   - sometimes $\hat{f}$ used instead of $h$

5 Data-generation model: instances are generated by some probability distribution and labeled according to a function
   - $\mathcal{D}$: probability distribution over $\mathcal{X}$ (NOT KNOWN TO THE LEARNER!)
   - labeling function $f : \mathcal{X} \to \mathcal{Y}$ (NOT KNOWN TO THE LEARNER!)
   - label $y_i$ of instance $x_i$: $y_i = f(x_i)$, for all $i = 1, \ldots, m$
   - each point in training set $S$: first sample $x_i$ according to $\mathcal{D}$, then label it as $y_i = f(x_i)$

6 Measures of success: error of a classifier = probability it does not predict the correct label on a random data point generated by distribution $\mathcal{D}$
Loss

Given domain subset $A \subset \mathcal{X}$, $\mathcal{D}(A)$ = probability of observing a point $x \in A$.

In many cases, we refer to $A$ as event and express it using a function $\pi : \mathcal{X} \rightarrow \{0, 1\}$, that is:

$$A = \{x \in \mathcal{X} : \pi(x) = 1\}$$

In this case we have $\mathbb{P}_{x \sim \mathcal{D}}[\pi(x)] = \mathcal{D}(A)$

Error of prediction rule $h : \mathcal{X} \rightarrow \mathcal{Y}$ is

$$L_{\mathcal{D},f}(h) \overset{\text{def}}{=} \mathbb{P}_{x \sim \mathcal{D}}[h(x) \neq f(x)] \overset{\text{def}}{=} \mathcal{D}(\{x : h(x) \neq f(x)\})$$

Notes:

- $L_{\mathcal{D},f}(h)$ has many different names: generalization error, true error, risk, loss, ...
- often $f$ is obvious, so omitted: $L_{\mathcal{D}}(h)$
Learning Process (Simplified)

**UNKNOWN TARGET FUNCTION**
\[ f : \mathcal{X} \rightarrow \mathcal{Y} \]

**PROBABILITY DISTRIBUTION**
\[ \mathcal{D} \text{ on } \mathcal{X} \]

**TRAINING EXAMPLES**
\[(x_1, y_1), \ldots, (x_m, y_m)\]

**LOSS FUNCTION**
\[ L() \]

**LEARNING ALGORITHM** \( \mathcal{A} \)

**FINAL HYPOTHESIS/MODEL**
\[ \hat{f} : \mathcal{X} \rightarrow \mathcal{Y} \]

\[ \hat{f}(x) \approx f(x) \]
Types of Learning

$y_i$ are known: training set $(x_1, y_1), \ldots, (x_m, y_m)$

supervised learning

$y_i$ are not known: training set $x_1, x_2, \ldots, x_m$

unsupervised learning

There can be different types of output:

- $Y$ is discrete
- $Y$ is continuous

Notes: we will see a more general learning model soon, main ideas are the same!
Types of Learning

\[ Y_i \text{ known} \quad Y_i \text{ not known} \]

**Supervised Learning**
- Classification
- Regression

**Unsupervised Learning**
- Clustering
- Dimensionality reduction
- ...

\[ Y \text{ is} \]

- Continuous
- Discrete
(Rough) Course Plan

PART I: Supervised Learning

Introduction: Data, Classes of models, Losses, Probabilistic models and assumptions on the data, Models, Losses

Regression and Classification

When is a model good? Model complexity, bias variance tradeoff/generalization (VC dimension generalization error)

Least Squares, Maximum Likelihood and Posteriors.

Models for Regression Linear Regression (scalar and multivariate), Regularization, Subset Selection

Linear-in-the-parameters models, Regularization.

Classes of non linear models: Sigmoids, Neural Networks

Kernel Methods: SVM
(Rough) Course Plan

Models for Classification: Logistic Regression, NN, Naive Bayes Classifier, SVM

Validation and Model Selection

Model complexity determination

PART II: Unsupervised learning

Cluster analysis: K-means Clustering, Mixtures of Gaussians and the EM estimation

Dimensionality reduction: Principal Component Analysis (PCA)

Part III: Advanced Topics

Deep Learning

Statistically Significant Patterns

...
Objectives

provide the fundamentals and basic principles of the learning problem

introduce the most common algorithms for regression and classification

analytical and practical ability in using these tools for the solution of basic problems

some hands-on experience
Course Prerequisites!

Calculus

Programming

Linear Algebra

Probability
Calculus

- derivatives
- minimization of functions
- partial derivatives of functions of multiple variables
- integrals
Programming

• You should already know Java
  ... learning Python will be easy!
Linear Algebra

- matrix factorization
- matrix inversion
- linear independence
- rank, column space, null space
- orthogonality, projections
- eigenvalues, eigenvectors
- symmetric positive definite matrices
- matrix differentiation
Probability

- discrete random variables (r.v.), moments, expectation
- continuous r.v.’s, probability density function (PDF), cumulative distribution function (CDF)
- joint, marginal, conditional distribution
- some famous distributions:
  - discrete: binomial
  - continuous: Gaussian
- Independence and conditional independence
- Bayes Theorem
- Law of large numbers