Introduction to the ISPR Course

Davide Bacciu

Dipartimento di Informatica
Università di Pisa
bacciu@di.unipi.it

Intelligent Systems for Pattern Recognition (ISPR)

Objectives

Train machine learning (ML) specialists capable of

- designing novel learning models
- developing pattern recognition applications using ML

Focus on challenging and complex data

- Machine Vision: noisy, hard-to-interpret, semantically rich information
- Structured data: relational information; sequences, trees, graphs

Lectures do not cover Natural Language Processing as there is a dedicated course

Expected Outcome

Methodology-oriented outcomes

- Gain in-depth knowledge of advanced machine learning models
- Understand the underlying theory
- Be able to individually read, understand and discuss research works in the field

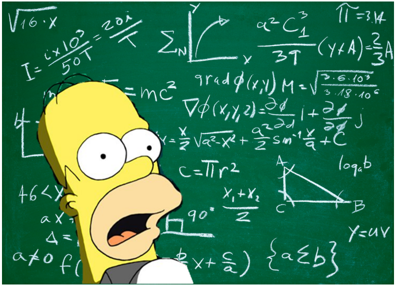
Application-oriented outcomes

- Learn to address modern pattern recognition problems
- Gain knowledge of ML and PR libraries
- Be able to develop an application using ML models

Prerequisites

- Knowledge of machine learning fundamentals
  - Pass the ML course or.. come discuss your ML skills with me
- Mathematical *tools* for ML
  - Algebra and calculus
  - Optimization
  - Probability and statistics
- Programming experience in Python (and Matlab)

...and, above all, a disposition not to get (easily) scared by math!



Organization

The course covers four themes

- Introduction to Pattern Recognition
- Generative (probabilistic) Models
- Deep Learning
- Applications and Software

An incremental approach: from old school pattern recognition to state of the art deep learning

Guest Lectures

Guest seminars by Italian and international researchers and Ph.D. students

- Lectures by Alessio Micheli on neural networks for graphs
- Lectures by Jan Gosphert (@uni-bielefeld) on deep learning for machine vision
- Practical lectures on deep learning frameworks (PyTorch, Keras, Tensorflow)
- Short seminars on hot research topics by Ph.D. students
  - Generative models for structures
  - Adversarial machine learning
  - Advanced memory-based networks
  - Applications to music generation, life sciences, ...

Topics (I)

- Introduction to Pattern Recognition
  - Introduction to signal processing
  - Introduction to image processing
- Generative Models
  - Graphical models
  - Hidden Markov Models
  - Markov Random Fields
  - Boltzmann machines
  - Bayesian learning and variational inference

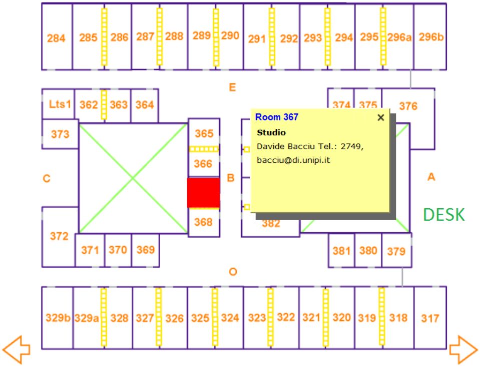
Topics (II)

- Deep Learning (DL) fundamentals
  - Convolutional architectures
  - Gated recurrent networks
  - Deep autoencoders
  - DL toolset: dropout, batch normalization, residual connections,
- Advanced learning models
  - Memory-enhanced networks
  - Generative deep learning
  - Deep reinforcement learning
- Applications in machine learning
  - Learning in structured domains
  - Machine vision, multimodal learning, BioInformatics, robotics,...

Course Instructor

Davide Bacciu

Email - `bacciu@di.unipi.it`
Tel - 050 2212749
Office - Room 367, Dipartimento di Informatica
Office hours - Thursday 16-18 (email me!)



Course Schedule

Weekly Timetable:

Day	Time	Room
Thursday	14-16	C1
Friday	11-13	L1

Talk now if you need to change course weekly schedule!

Course comprises 24 lectures

- Need two dates to recover the 14-15 March lessons (likely April)
- Will need to accommodate some (2,3) extra dates for midterms

Course Homepage

Reference Webpage on Moodle:

`elearning.di.unipi.it/course/view.php?id=110`

Here you can find

- Course information
- Lecture slides Maybe recorded lectures
- Articles and course materials
- Midterm and final project assignments

Subscribe to the course to receive feeds and news

Reference Books

No official textbook

Generative learning reference (free pdf, with code):

David Barber, *Bayesian Reasoning and Machine Learning*, Cambridge University Press (2012)

Deep learning reference (free pdf):

I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, MIT Press (2016)

For pattern recognition refer to slides (and additional material)

The Origins of Pattern Recognition (PR)

Duda and Hart, 1973

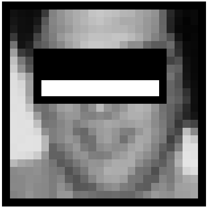
Machine recognition of meaningful regularities in noisy or complex environments

A variety of approaches to realize it

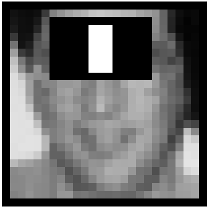
- Statistical PR
- Clustering
- Rule-based systems (fuzzy)
- Signal processing
- Logic and reasoning
- Structural and syntactic PR
- ...and of course machine learning!

The Viola-Jones Algorithm

Consider the following two hand drawn pixel masks



VJ1



VJ2

Sum pixels in the white area and subtract those in the black portion

- VJ1 is large in the eye region
- VJ2 is large on the nose stripe

VJ algorithm positions the masks on the image and combines the responses ($\approx 5K$ hand aligned examples)

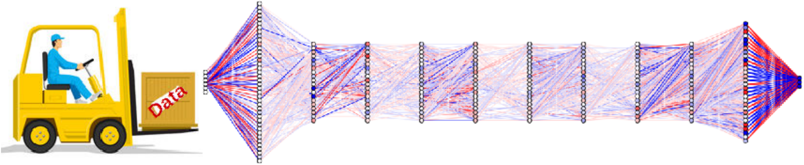
PR Stages - An historical View

1. Identification of distinguishing attributes of the object/entity (feature detection)
2. Extraction of features for the defining attributes (feature extraction)
3. Comparison with known patterns (matching)

Basically, lots of sleepless nights hand-engineering the best data features

PR Stages - A Modern View

Pattern recognition after the deep learning revolution



Apparently a single stage process with a data crushing-and-munching neural monster spitting out predictions

Modern Pattern Recognition

Presentation continues out of here

The Course Philosophy

- Start from traditional PR approaches
  - Introduce problems and tasks
  - Learn some useful techniques
- Learn how old-school stuff has been reused in a modern way
- Understand how traditional PR relates to recent advances

A practical approach with code complementing theory when possible

Reference Languages

Reference languages for the course are Python and (some) Matlab

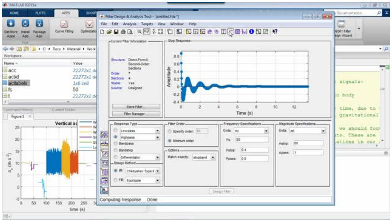
- Students of the AI curriculum should be already familiar with both
- Easy-to-learn languages enhanced by reasonable editors and graphical environments
- Lots of library support for signal processing, image processing and machine learning

For the final project there is some reasonable flexibility in which language you can use (no deep learning in Pascal, please!)

Why Matlab

- Excellent for linear algebra
- Decent GPU support (`gpuarray` and you are done)
- Loads of algorithms and functionalities for
  - Signal processing
  - Image analysis

Graphical editor and development environment slightly better than Python



Reasonable for very quick and dirty prototyping of non-neural models

Why Python

- More fully-fledged programming language
- Support for vectorization and GPU (at the price of some swearing at installation time)
- Loads of useful libraries for
  - Machine learning
  - Deep learning
  - Machine vision

The reference language for machine learning

Keras
TensorFlow
scikit learn
PYTORCH

A must if you want the AI community to accept and use your model

Useful Python Modules

- numpy - Matrix / numerical analysis layer
- scipy - Scientific computing utilities: linear algebra, signal/image processing, ...
- pandas - Data wrangling
- matplotlib - Plotting and visualization
- opencv - Computer vision
- scikit-learn - Machine learning
- statsmodels - Statistics in Python
- Tensorflow, Keras, Pytorch - Deep learning

Python Tips

1. Windows users: get Anaconda
2. Get an IDE: e.g. PyCharm or Spyder (with Anaconda)
3. Set up VirtualEnv: configure once, port everywhere
4. Want to use GPU with fancy Deep Learning libraries?
    - Consider using docker
    - Different libraries have different CUDA-CuDNN support

Jupiter notebooks are a good way to interactively experiment with data in a Matlab-like fashion

Exams

Student following the course lecture can complete the exam as a 3-step process

- **Midterm Assignignments** - A total of 3 short presentations (5 minutes) on experiences related to course topics
- **Final Project** - Presentation slides or a documented software on a topic of interest for the course (and for you)
- **Oral Exam** - A presentation of the final project plus examination on the course program

The alternative way (for working students, those not attending classes and those who fail the other way)

- **Final Project** - A written report AND a software on a topic of interest for the course
- **Oral Exam** - A presentation of the final project plus examination on the course program

Midterm Assignments

- A very short presentation (5 minutes) to be given in front of the class on one of the following experiences
  - A quick and dirty (but working) implementation of a simple pattern recognition algorithm
  - A report concerning the experience of installing and running a demo application realized using available libraries
  - A summary of a recent research paper on topics/models related to the course content.
- The presenter should be able to answer my (and your collegues') questions on the presentation
- Timeline
  - One midterm per month
  - Midterm published: late February, late March, early May
  - Midterm discussion: late March, late April, late May

Final Project (I)

- Choose from a set of suggested topics or propose your own topic of interest
- Timeline (quick way)
  - Suggested topics list: early-may
  - Choose project: strictly before the last lecture (late may)
  - Presentation delivery: by the standard exam date (appello) (strict)
- Timeline (alternative way)
  - Choose project: email me to arrange a topic
  - Report and presentation delivery (6-10 pages): by the standard exam date (appello) (strict)

Final Project (II)

- Possible project types

  Survey Read at least three relevant and distinct papers on a topic and prepare a presentation (or write a report): not a simple summary, rather try to find connections between the works and highlight interesting open problems

  Software Develop a well-written, tested and commented software implementing a non-trivial learning model and/or a pattern recognition application relevant for the course

Oral Exam

- Give your presentation (20 minutes) on the final project
  - Discuss it in front of me and anybody interested
  - Be prepared to answer my questions on the presentation
- After the presentation candidates will be subject to an oral exam with questions covering the course contents
- Remember to send the presentation by the appello deadline (will also create a submission on Moodle)

How to get past this course?

Grading (with midterms) $G = G_P + G_O + \sum_{i=1}^{3} G_M^i$

- $G_P \in [1, 15]$ is the final project grade
- $G_O \in [1, 14]$ is the oral grade
- $G_M^i \in \{0, 1\}$ (sometimes $\{0, 1, 2\}$) is the increment for the i-th midterm

Grading (alternative way) $\frac{(G_P + G_O)}{2}$

- $G_P \in [1, 32]$ is the project grade
- $G_O \in [1, 30]$ is the oral grade

Upcoming..

Introduction to Pattern Recognition

An introduction to the fundamental PR problems in signal and image processing and a summary of the old-school techniques to address them.

Topics

- Timeseries analysis
- Convolution and correlation operators
- Visual feature descriptors
- Visual feature detectors
- Image segmentation

Next Lecture

Introduction to Signal Processing

- Timeseries item Convolution and correlation
- Spectral analysis

Showcooking

Whenever possible theoretical aspects will be complemented with coding examples

Contacts and Information

Remember to register on the course Moodle

`elearning.di.unipi.it/course/view.php?id=110`

Within the end of this week please send a mail with your email address for the course mailing list

- Object should include tag [ISPR] (or may end up in thrash)
- Put your name, email and curriculum/course in the body

Questions?