MACHINE LEARNING

Liviu Ciortuz
Department of CS, University of Iași, România
What is Machine Learning?

- ML studies algorithms that improve with experience.

Tom Mitchell’s Definition of the [general] learning problem:
“A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance on tasks in T, as measured by P, improves with experience E.”

- Examples of [specific] learning problems (see next slide)
- [Liviu Ciortuz:] ML is data-driven programming
- [Liviu Ciortuz:] ML gathers a number of well-defined sub-domains/disciplines, each one of them aiming to solve in its own way the above-formulated [general] learning problem.
What is Machine Learning good for?

- natural language (text & speech) processing
- genetic sequence analysis
- robotics
- customer (financial risk) evaluation
- terrorist threat detection
- compiler optimisation
- semantic web
- computer security
- software engineering
- computer vision (image processing)
- etc.
A multi-domain view

- Artificial Intelligence (concept learning)
- Statistics (model fitting)
- Machine Learning
- Data Mining
- Pattern Recognition
- Algorithms
- Mathematics
- Database Systems (Knowledge Discovery in Databases)
- Engineering
The Machine Learning Undergraduate Course: Plan

0. Introduction to Machine Learning (T. Mitchell, ch. 1)

1. Probabilities Revision (Ch. Manning & H. Schütze, ch. 2)

2. Decision Trees (T. Mitchell, ch. 3)

3. Parameter estimation for probablistic distributions
   (see Estimating Probabilities, additional chapter to T. Mitchell’s book, 2016)

4. Bayesian Learning (T. Mitchell, ch. 6)
   and the relationship with Logistic Regression

5. Instance-based Learning (T. Mitchell, ch. 8)

6. Clustering Algorithms (Ch. Manning & H. Schütze, ch. 14)
The Machine Learning Master Course:  
Tentative Plan

1. Probabilities Revision (Ch. Manning & H. Schütze, ch. 2)  
2. Decision Trees: Boosting  
3. Gaussian Bayesian Learning  
4. The EM algorithmic schemata (T. Mitchell, ch. 6.12)  

6. Hidden Markov Models (Ch. Manning & H. Schütze, ch. 9)  
7. Computational Learning Theory (T. Mitchell, ch. 7)
Bibliography

0. “Exerciții de învățare automată”
   L. Ciortuz, A. Munteanu E. Bădărău.
   Editura Universității “Alexandru Ioan Cuza”, Iași, Romania, 2018

1. “Machine Learning”

2. “The Elements of Statistical Learning”
   Trevor Hastie, Robert Tibshirani, Jerome Friedman. Springer, 2nd ed. 2009

3. “Machine Learning – A Probabilistic Perspective”
   Kevin Murphy, MIT Press, 2012

4. “Pattern Recognition and Machine Learning”
   Christopher Bishop. Springer, 2006

5. “Foundations of Statistical Natural Language Processing”
   Christopher Manning, Hinrich Schütze. MIT Press, 2002
Other suggested readings:
More on the theoretical side (I)

1. “Pattern Recognition” (2nd ed.)
2. “Bayesian Reasoning and Machine Learning”
   David Barber, 2012
5. “Apprentissage artificiel” (2e ed.)
   Antoine Cornuéjols. Eyrolles, 2010
Other suggested readings:
More on the theoretical side (II)

1. “Data mining with decision trees” (2nd ed.)
2. “Clustering”
   Rui Wu, Donald C. Wunsch II; IEEE Press, 2009

3. “The EM Algorithm and Extensions” (2nd ed.)
4. “A Tutorial on Support Vector Machines for Pattern Recognition”
   Christopher Burges, 1998
5. “Support Vector Machines and other kernel-based learning methods”

6. “Apprentissage statistique. Réseaux de neurones, cartes topologiques, machines à vecteurs supports” (3e ed.)
Other suggested readings: More on the practical side


2. “An Introduction to Statistical Learning”
   Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. Springer, 2013

   Max Kuhn, Kjell Johnson; Springer, 2013

   Sergios Theodoridis, Konstantinos Koutroumbas. Academic Press, 2010


6. “Data Mining with R – Learning with Case Studies”
   Luís Torgo. CRC Press, 2011

7. “Mining of Massive Datasets”
   Anand Rajaraman, Jure Leskovec, Jeffrey D. Ullman; 2013
A general schema for machine learning methods

“We are drowning in information but starved for knowledge.”
Basic ML Terminology

1. instance $x$, instance set $X$
   concept $c \subseteq X$, or $c : X \rightarrow \{0, 1\}$
   example (labeled instance): $(x, c(x))$; positive examples, neg. examples

2. hypotheses $h : X \rightarrow \{0, 1\}$
   hypotheses representation language
   hypotheses set $H$
   hypotheses consistent with the concept $c$: $h(x) = c(x), \forall$ example $(x, c(x))$
   version space

3. learning = train + test
   supervised learning (classification), unsupervised learning (clustering)

4. $error_h = | \{ x \in X, h(x) \neq c(x) \} |$
   training error, test error
   accuracy, precision, recall

5. validation set, development set
   $n$-fold cross-validation, leave-one-out cross-validation
   overfitting
The Inductive Learning Assumption

Any hypothesis found to conveniently approximate the target function over a sufficiently large set of training examples will also conveniently approximate the target function over other unobserved examples.
Inductive Bias

Consider

- a concept learning algorithm $L$
- the instances $X$, and the target concept $c$
- the training examples $D_c = \{\langle x, c(x) \rangle \}$.
- Let $L(x_i, D_c)$ denote the classification assigned to the instance $x_i$ by $L$ after training on data $D_c$.

**Definition:**

The inductive bias of $L$ is any minimal set of assertions $B$ such that

$$(\forall x_i \in X)[(B \lor D_c \lor x_i) \vdash L(x_i, D_c)]$$

for any target concept $c$ and corresponding training examples $D_c$.

($A \vdash B$ means $A$ logically entails $B$)
Inductive systems can be modelled by equivalent deductive systems.
Evaluation measures in Machine Learning

- **accuracy**: \( \text{Acc} = \frac{tp + tn}{tp + tn + fp + fn} \)
- **precision**: \( P = \frac{tp}{tp + fp} \)
- **recall (or: sensitivity)**: \( R = \frac{tp}{tp + fn} \)
- **F-measure**: \( F = \frac{2 P \times R}{P + R} \)
- **specificity**: \( Sp = \frac{tn}{tn + fp} \)
- **follout**: \( \frac{fp}{tn + fp} \)

Mathew’s Correlation Coefficient:

\[
MCC = \frac{tp \times tn - fp \times fn}{\sqrt{(tp + fp) \times (tn + fn) \times (tp + fn) \times (tn + fp)}}
\]
Lazy learning vs. eager learning algorithms

**Eager:** generalize before seeing query

- ID3, Backpropagation, Naive Bayes, Radial basis function networks, ...
- Must create global approximation

**Lazy:** wait for query before generalizing

- $k$-Nearest Neighbor, Locally weighted regression, Case based reasoning
- Can create many local approximations

Does it matter?

If they use the same hypothesis space $H$, lazy learners can represent more complex functions.

E.g., a lazy Backpropagation algorithm can learn a NN which is different for each query point, compared to the eager version of Backpropagation.
Who is Liviu Ciortuz?

- Diploma (maths and CS) from UAIC, Iași, Romania, 1985
- PhD in CS from Université de Lille, France, 1996
- programmer:
  Bacău, Romania (1985-1987)
- full-time researcher:
  Germany (DFKI, Saarbrücken, 1997-2001),
  UK (Univ. of York and Univ. of Aberystwyth, 2001-2003),
  France (INRIA, Rennes, 2012-2013)
- assistant, lecturer and then associate professor:
ADDENDA

“...colleagues at the Computer Science department at Saarland University have a strong conviction, that nothing is as practical as a good theory.”

Reinhard Wilhelm, quoted by Cristian Calude, in *The Human Face of Computing*, Imperial College Press, 2016
“Mathematics translates concepts into formalisms and applies those formalisms to derive insights that are usually NOT amenable to a LESS formal analysis.”

“Mathematics is a journey that must be shared, and by sharing our own journey with others, we, together, can change the world.”

“Through the power of mathematics, we can explore the uncertain, the counterintuitive, the invisible; we can reveal order and beauty, and at times transform theories into practical objects, things or solutions that you can feel, touch or use.”

Cedric Villani, winner of the Fields prize, 2010

ADMINISTRATIVIA
Grading standards for the ML course

Obiectiv: invatare pe tot parcursul semestrului!

Punctaj

<table>
<thead>
<tr>
<th>S1</th>
<th>P1</th>
<th>S2</th>
<th>P2</th>
<th>Extras:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seminar</td>
<td>Partial 1</td>
<td>Seminar</td>
<td>Partial 2</td>
<td>Seminar special</td>
</tr>
<tr>
<td>6p</td>
<td>12p</td>
<td>6p</td>
<td>12p</td>
<td>...</td>
</tr>
<tr>
<td>Minim: 2p + 4p</td>
<td>Minim: 2p + 4p</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Prezenta la seminar: obligatorie!
Penalizare: 0.1p pentru fiecare absenta de la a doua incolo

Nota = \((4 + S1 + P1 + S2 + P2) / 4\)
Pentru promovare: \(S1 + P1 + S2 + P2 \geq 14\)
REGULI generale pentru cursul de Învățare automată (cont.)

Sistemul de notare

Nota = (4 + S1 + P1 + S2 + P2) / 4,
unde

- $S_1$ = punctajul la seminar pe prima jumătate de semestru (0-6 puncte)
- $S_2$ = punctajul la seminar pe a doua jumătate de semestru (0-6 puncte)
- $P_1$ = punctajul la primul examen parțial (0-12 puncte)
- $P_2$ = punctajul la al doilea examen parțial (0-12 puncte)

Punctajele $S_1$ și $S_2$ se obțin (fiecare) ca medie aritmetică a două punctaje, pentru
- răspunsuri “la tablă”
- test scris (anunțat în prealabil)

Condiții de promovare:
$S_1 \geq 2; S_2 \geq 2; P_1 \geq 4, P_2 \geq 4$, nota $\geq 4.5$
În consecință, punctajul minimal de îndeplinit din suma $S_1+P_1+S_2+P_2$ este 14.

Atenție:
$S_1 < 2$ (sau $S_2 < 2$) implică imediat nepromovarea acestui curs în anul universitar 2018–2019!
REGULI generale pentru cursul de Învățare automată (cont.)
pentru cursul de la licență

- **Slide-uri de imprimat** (în această ordine și, de preferat, COLOR):
  

  (Atenție: acest set de slide-uri poate fi actualizat pe parcursul semestrului!)

- **De imprimat (ALB-NEGRU):**
  
  - http://profs.info.uaic.ro/~ciortuz/SLIDES/ml0.pdf
REGULI generale pentru cursul de Învățare automată (cont.) pentru cursul de la master

- **Slide-uri de imprimat** (în această ordine și, de preferat, COLOR):
  
  (Atenție: acest set de slide-uri poate fi actualizat pe parcursul semestrului!)

- De imprimat (ALB-NEGRU):
  

- De imprimat opțional (ALB-NEGRU):
  **Companion-ul practic** pentru culegerea „Exerciții de învățare automată“:
REGULI generale pentru cursul de Învățare automată (cont.)

Observație (1)
La fiecare curs și seminar, studenții vor veni cu cartea de exerciții și probleme (de L. Ciortuz et al) și cu o fasciculă conținând slide-urile imprimate.

Observație (2)
Profesorul responsabil pentru acest curs, Liviu Ciortuz NU va răspunde la email-uri care pun întrebări pentru care răspunsul a fost deja dat
– fie în aceste slide-uri,
– fie la curs