

MACHINE LEARNING

2003–2009

Liviu Ciortuz

Department of CS, University of Iași, România

What is Machine Learning?

- ML studies the programs that improve with experience
- [Liviu Ciortuz:] ML is data-driven programming
- [some people:] “When nothing else works, try ML!”

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.

The Machine Learning 2008 Course: Tentative Plan

1. Decision Trees (T.Mitchell, ch.3)
2. Instance-based Learning (T.Mitchell, ch.8)
3. Bayesian Learning (T.Mitchell, ch.6)
4. Support Vector Machines (N.Cristianini & J.Shawe-Taylor, 2000)

5. Clustering Algorithms (Ch.Manning & H.Schütze, ch.14)

6. Evaluating Hypotheses (T.Mitchell, ch.5)
7. Concept Learning and General-to-Specific Ordering (T.Mitchell, ch.2)
8. Computational Learning Theory (T.Mitchell, ch.7)

9. Boosting, Bagging and Random Forests

Bibliography

1. “Machine Learning”
Tom Mitchell; McGraw-Hill, 1997
 2. “Support Vector Machines and other kernel-based learning methods”
Nello Cristianini, John Shawe-Taylor, Cambridge University Press, 2000.
-
3. “Data Mining with SQL Server 2005”,
ZhaoHui Tang, Jamie MacLennan, Wiley Publishing, 2005
 4. “Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations”
Ian Witten, Eibe Frank (2nd ed.), Morgan Kaufmann Publishers, 2005

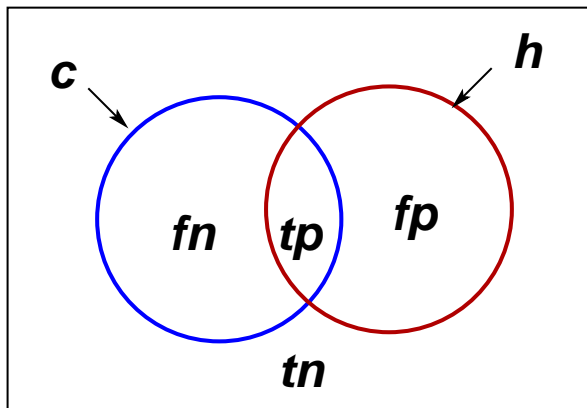
Additional Bibliography

1. “The Elements of Statistical Learning”
Friedman, Hastie, Tibshirani; Springer, 2001
2. “Pattern Recognition” (2nd ed.)
Duda, Hart, Stork; John Wiley & Sons Inc., 2001
3. “Pattern Recognition and Machine Learning”
Christopher Bishop; Springer, 2006
4. “Foundations of Statistical Natural Language Processing”
Christopher Manning, Hinrich Schütze; MIT Press, 2002
5. “Data Mining”,
Tan, Steinbach, Kumar; Pearson Education, 2006
6. “Principles of Data Mining (Adaptive Computation and Machine Learning)”,
Hang, Mannila, Smyth; MIT Press, 2001
7. “Advances in Machine Learning Applications in Software Engineering”,
Du Zhang, Jeffrey Tsai; Idea Group Publishing, 2007

ML Basic Terminology

1. instance x , instance set X
concept $c \subseteq X$, or $c : X \rightarrow \{0, 1\}$
(labeled) example: $\langle x, c(x) \rangle$, positive examples, negative 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 $\langle x, c(x) \rangle$
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
overfitting, cross-validation, leave-one-out cross-validation

Evaluation measures in Machine Learning



tp – true positives
 fp – false positives
 tn – true negatives
 fn – false negatives

accuracy: $Acc = \frac{tp + tn}{tp + tn + fp + fn}$

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:

$$MCC = \frac{tp \times tn - fp \times fn}{\sqrt{(tp + fp) \times (tn + fn) \times (tp + fn) \times (tn + fp)}}$$

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

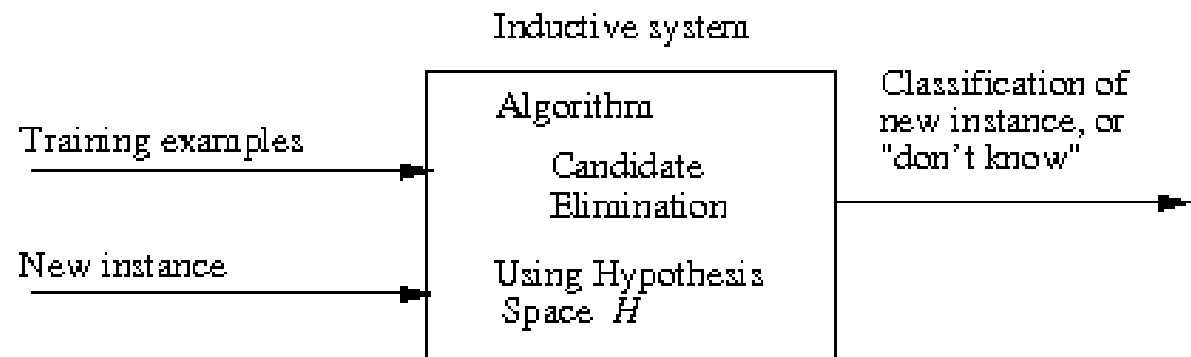
- concept learning algorithm L
- instances X , target concept c
- training examples $D_c = \{\langle x, c(x) \rangle\}$
- let $L(x_i, D_c)$ denote the classification assigned to the instance x_i by L after training on data D_c .

Definition:

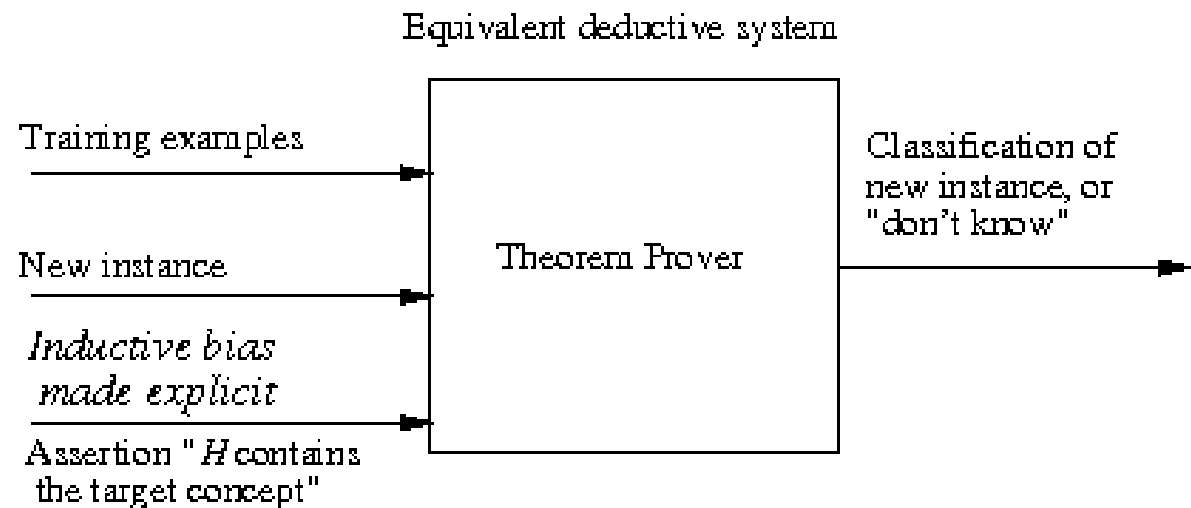
The **inductive bias** of L is any minimal set of assertions B such that

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

for any target concept c and corresponding training examples D_c



**Inductive systems
can be modelled by
equivalent deductive
systems**



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.