Artificial Intelligence: Everything You Need to Know in 90 Minutes

Mark Maloof

Department of Computer Science
Georgetown University
Washington, DC 20057-1232
http://www.cs.georgetown.edu/~maloof

Governing Emerging Technologies (CCTP-779)

5 October 2015

Outline

- What is AI?
- Computation, Algorithms, Turing Machines, and Limits
- Logic, Probability, Numbers, and Logic and Probability
- Hypercomputation (and pseudo-hyper computation!)
- Stanley: A reason to be optimistic
- What about me!?
- Bring it on home

What is AI?

McCarthy et al., 1955

- "The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it."

Haugeland, 1985

- "The exciting new effort to make computers think...machines with minds, in the full and literal sense."

Charniak and McDermott, 1985

- "...the study of mental faculties through the use of computational models."

Rich and Knight, 1992, 2009

- "The study of how to make computers do things at which, at the moment, people are better."

Nilsson, 1998

- "Artificial intelligence, broadly (and somewhat circularly) defined, is concerned with intelligent behavior in artifacts. Intelligent behavior, in turn, involves perception, reasoning, learning, communicating, and acting in complex environments."

Russell and Norvig's Four Approaches

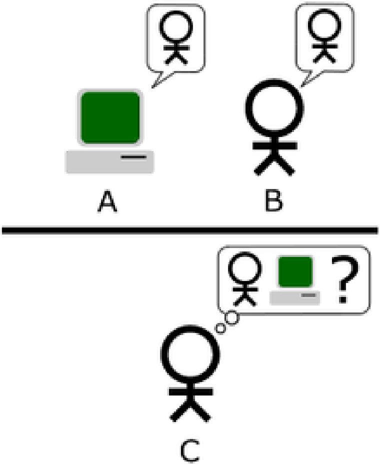
1. Think like a human
2. Act like a human
3. Think rationally
4. Act rationally

Think Like A Human

- "...machines with minds, in the full and literal sense"
- Put simply, program computers to do what the brain does
- How do humans think?
- What is thinking, intelligence, consciousness?
- If we knew, can computers do it, think like humans?
- Does the substrate matter, silicon versus meat?
- Computers and brains have completely different architectures
- Is the brain carrying out computation?
- If not, then what is it?
- Can we know ourselves well enough to produce intelligent computers?

Act Like A Human

Turing Test



Source: http://en.wikipedia.org/wiki/Turing_test

Obligatory xkcd Comic

Source: http://xkcd.com/329/

The Brilliance of the Turing Test

- Sidesteps the hard questions:
  - What is intelligence?
  - What is thinking?
  - What is consciousness?
- If humans can't tell the difference between human intelligence and artificial intelligence, then that's it
- Proposed in 1950, Turing's Imitation Game is still relevant

Think Rationally

- Think rationally? Think logic!
- Put simply, write computer programs that carry out logical reasoning
  - Logic: propositional, first-order, modal, temporal, ...
  - Reasoning: deduction, induction, abduction, ...
- Possible problem: Humans don't really think logically
- Do we care? Strong versus weak AI
- One problem: often difficult to establish the truth or falsity of premises
- Another: conclusions aren't strictly true or false

Act Rationally

- Act rationally? Think probability and decision theory!
- "A rational agent is one that acts so as to achieve the best outcome or, when there is uncertainty, the best expected outcome" (Russell and Norvig, 2010, p. 4)
- <jab>"when there is uncertainty"</jab>
- When *isn't* there uncertainty?
- Predominant approach to AI (for now)

Computation!

Binary

- Everything in a computer is binary: 0 or 1
- Start with one wire and two voltage levels:
  - 0–2 volts $\Rightarrow$ 0
  - 3–5 volts $\Rightarrow$ 1
- One wire $\equiv$ one binary digit $\equiv$ one bit
- What can you do?
  - change 0 to 1
  - change 1 to 0
- This state change is computation at its most basic level
- Not very interesting, but wait! There's more!

Bits, Bytes, and Gigs

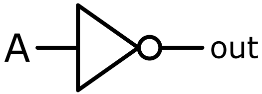
- Why limit ourselves to one wire?
  - One wire $\equiv$ bit
  - Eight wires $\equiv$ byte, $2^8 = 256$
- Most computers and smartphones are 64-bit devices
  - $2^{64} = 18,446,744,073,709,551,616$
- Implications:
  - What the device can store at each memory location (word size)
  - How many memory locations the device can address (address size)

Binary Representations of Stuff

- Everything in a computer is coded in a binary representation
  - integers: $7 \equiv 0100$
  - real numbers $-2.7 \equiv -27 \times 10^{-1} \equiv 111011100001$
    - $-27 \times 10^{-1} \equiv 1\ 11011\ 1\ 00001$
  - characters 'A' $\equiv 65 \equiv 01000001$
  - sequence of characters
  - color: integers for red, blue, green intensity
  - images: two-dimensional organization of colors
  - video: sequence of images
- Critical point: Every operation is a numeric computation, even if it involves changing your last name.

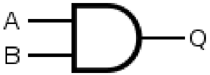
NOT

A	out
0	1
1	0

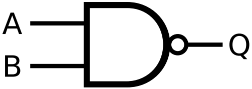


AND

A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1



Computation: Beautiful NAND



inputs		output
A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

NAND is Cool!

- NAND is *functionally complete*
  - Anything computable can be computed using only NAND gates
- This is not controversial
- It's descriptive, but it's not constructive
  - Tells you *that*, but not *how*
- So is the brain carrying out computation?
- That's the difficult question
- You can't just answer no
- You have to explain that not-computation process
- That's even more difficult

What is *Computation*?

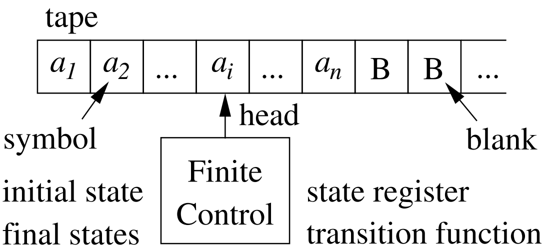
- "The 'computable' numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means" (Turing, 1936)
- Therefore, computation is the means of calculating such computable numbers
- What is a computable number?
- What is not a computable number?

Turing Machines

Basic Ideas

- A Turing machine is a mathematical model of computation
- Turing machines can compute anything
- They can produce any computable number

Turing Machines
Pictorially



Source: Hopcroft and Ullman (1979, Fig. 7.1, p. 148)

Turing Machines

Formally

- Formally, following Hopcroft and Ullman (1979), a Turing machine is the 7-tuple

$$M = \langle Q, \Gamma, B, \Sigma, \delta, q_0, B, F \rangle \ ,$$

  where
  - Q is the finite set of *states*,
  - Γ is the *alphabet*, a finite set of allowable tape symbols,
  - B is the blank symbol, which is included in Γ,
  - Σ is the set of *input symbols*, which is a subset of Γ excluding B,
  - δ is the *transition function*, which is a mapping from $Q \times \Gamma$ to $Q \times \Gamma \times \{L, R\}$,
  - q_0 is the *start state* and is in Q,
  - F is the set of *final states* and is a subset of Q.

Turing Machines

TM for AND

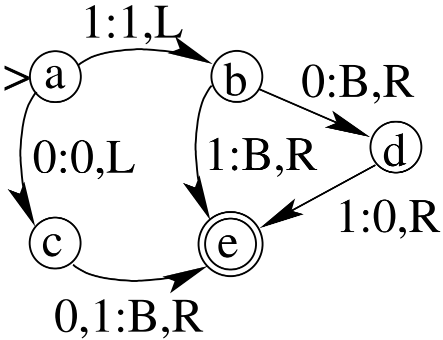
- States: $Q = \{a, b, c, d, e\}$
- Alphabeta: $\Gamma = \{0, 1, B\}$
- Input symbols: $\Sigma = \{0, 1\}$
- Transition function: $\delta =$

	symbol	
state	*0*	*1*
a	$c,\ 0,\ L$	$b,\ 1,\ L$
b	$d,\ B,\ R$	$e,\ B,\ R$
c	$e,\ B,\ R$	$e,\ B,\ R$
d	—	$e,\ 0,\ R$
e	—	—

- Start state: $q_0 = a$
- Final states: $F = \{e\}$

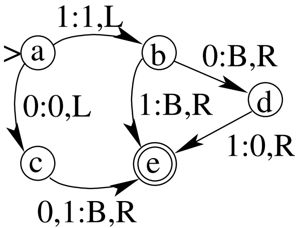
Turing Machines

TM State Diagram for AND



Turing Machines

TM Computation of AND

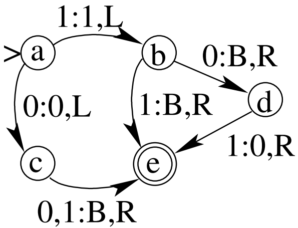


- Computation (1 AND 1 = 1):

$$\begin{aligned} a11 &\rightarrow 1b1 \\ &\rightarrow e1B \end{aligned}$$

Turing Machines

TM Computation of AND

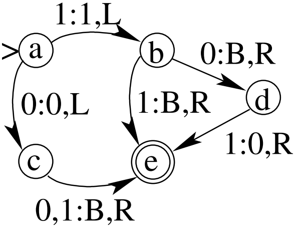


- Computation (0 AND 1 = 0):

$$\begin{aligned} a01 &\rightarrow 0c1 \\ &\rightarrow e0B \end{aligned}$$

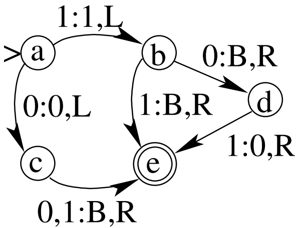
Turing Machines

TM Computation of AND



- Computation (1 AND 0 = 0):

$$\begin{aligned} a10 &\rightarrow 1b0 \\ &\rightarrow d1B \\ &\rightarrow e0B \end{aligned}$$



- Computation (0 AND 0 = 0):

$$\begin{aligned} a00 &\rightarrow 0c0 \\ &\rightarrow e0B \end{aligned}$$

Algorithms

- An algorithm is a formal, step-by-step process for accomplishing a computational task in a finite amount of time
- Critical elements:
  - Statements
  - Sequences of statements
  - Select whether to execute a sequence
  - Loop over a sequence until satisfying some condition

The Church-Turing Thesis

- Algorithms and Turing machines are equivalent

What about Programs?

- Turing Machines $\equiv$ Algorithms $\approx$ Programs $\equiv$ NAND Gates
- Why?
  - Programs are implementations of algorithms
  - Turing machines have infinite memory
  - Computers have finite memory (word size, address space)
  - Finite-length binary numbers only approximate real numbers
  - Therefore, computers approximate Turing machines
  - Does this matter? Probably not.
  - Computers are getting damn powerful
  - By the way, brains are also finite...

Uncomputability or Undecidability

- Perhaps a bigger problem: Not everything is computable
- That is, there are limits on what is computable
- Problems are decidable, undecidable, and semi-decidable
  - decidable: for all possible inputs, an algorithm exists that returns success or failure
  - undecidable: for all possible inputs, no algorithm exists that always returns success or failure
  - semi-decidable: more complicated, but let's say it's an algorithm that always returns success and either returns failure or does not halt (i.e., loops forever)
- Gödel, Post, Church, Turing, and others

The Halting Problem

- Formal algorithms for decidable problems halt and return either success or failure
- The Halting Problem:
  - There can be no algorithm that takes as input any other algorithm and returns success if the input algorithm halts
  - There can be no Turing machine that takes as input any other Turing machine and returns success if the input Turing machines halts

Some Big Questions

- Is intelligence a computable function?
  - Put another way: Is the brain a Turing machine?
  - Put another way: Can we **duplicate** the brain's processing?
  - What are the implications of the Halting Problem or is it a red herring?
- Are we relegated to **simulate** some or all of the brain's function?
  - The answer for 'some' is un-controversially yes.
  - All? Maybe.
- If the brain is not computing, what is it doing?
- Is the "binary approximation" of the world a problem?

Hypercomputation

- "The new field of hypercomputation studies models of computation that can compute more than the Turing machine and addresses their implications" (Ord, 2002)
- Computers $\approx$ Turing machines $<$ Hypercomputers
- On the other hand, "...there is no such discipline as hypercomputation" (Davis, 2006)
- Furthermore, Turing was not an idiot

Hypercomputation in a Nutshell

- Computers and Turing machines are digital (i.e., binary)
- The brain is analog (i.e., continuous)
  - what about spike trains?
- Digital is only an approximation to analog
  - yeah, but, sampling theorems!
- *Approximation* matters for some people
  - are we watching reality or just a movie?
  - is it chicken or does it taste like chicken?
- For some, *approximation* means Turing machines can't be minds
- Perhaps a device carrying out hypercomputation could
- But there are not yet any sufficiently powerful hypercomputers
- ...except, of course, the brain
- That is, brains perform hypercomputation; Turing machines can not; therefore, Turing machines can not be minds

Artificial Intelligence

Representation			
Reasoning			
Learning			

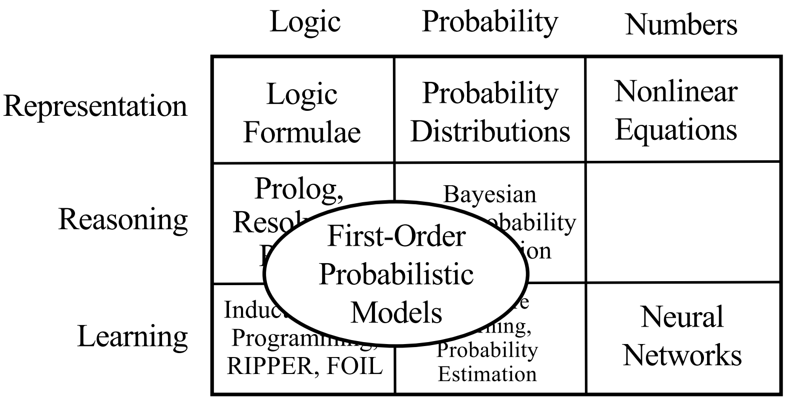
Artificial Intelligence

	Logic		
Representation	Logic Formulae		
Reasoning	Prolog, Resolution Proofs		
Learning	Inductive Logic Programming, RIPPER, FOIL		

Artificial Intelligence

	Logic	Probability	Numbers
Representation	Logic Formulae	Probability Distributions	Nonlinear Equations
Reasoning	Prolog, Resolution Proofs	Bayesian Nets, Probability Propagation	
Learning	Inductive Logic Programming, RIPPER, FOIL	Structure Learning, Probability Estimation	Neural Networks

Artificial Intelligence

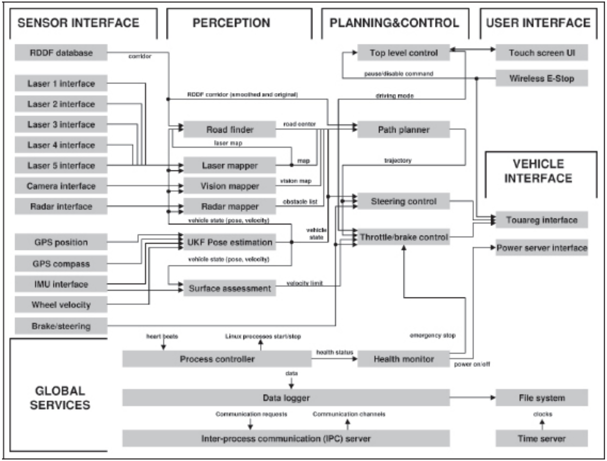


Stanley: A Reason to be Optimistic

- A self-driving car, a precursor to Google's self-driving car
- In 2005, drove a 175-mile course in the Mojave Desert
- Unaided by humans, who had only two-hours prior notice of the route
- Stanley used terrain maps to plan its overall route
- As it drove, it relied on its own analysis of "analytical relations and truths" to anticipate what lay ahead, by navigating the road itself, assessing its condition, and avoiding obstacles

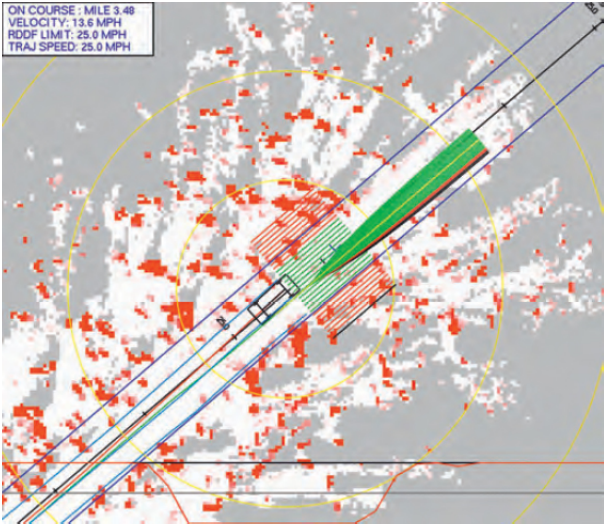
Video: The Great Robot Race

Stanley



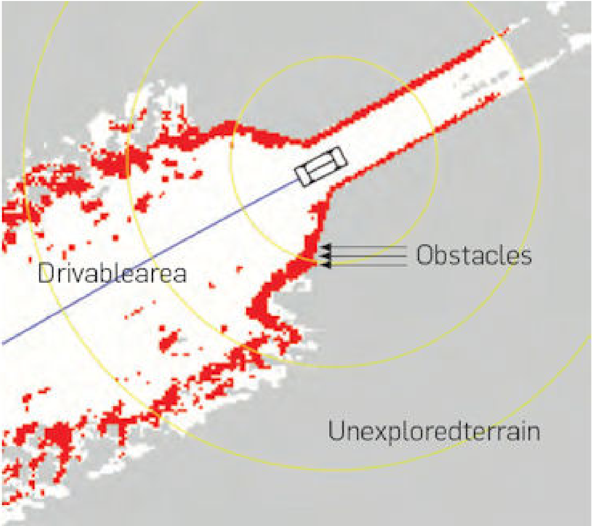
Source: Thrun (2010, Figure 2)

Stanley



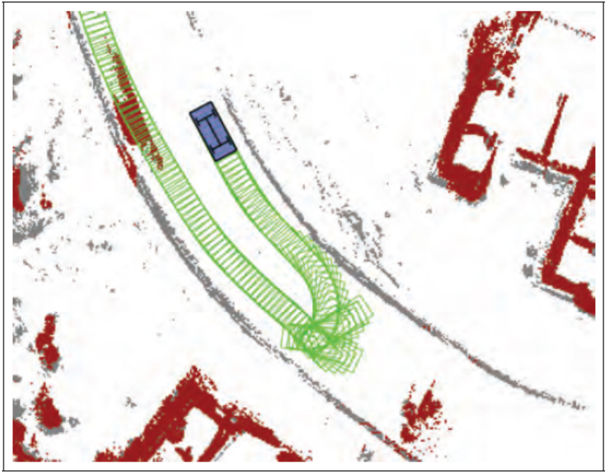
Source: Thrun (2010, Figure 7)

Stanley



Source: Thrun (2010, Figure 9a)

Stanley



Source: Thrun (2010, Figure 13)

What about Me?

	Logic	Probability	Numbers
Representation	Logic Formulae	Probability Distributions	Nonlinear Equations
Reasoning	Prolog, Resolution Proofs	Bayesian Nets, Probability Propagation	
Learning	Inductive Logic Programming, RIPPER, FOIL	Structure Learning, Probability Estimation	Neural Networks

Machine Learning

Or Pick Your Favorite Term...

- Given data, develop or use computational methods to build models that
  1. predict something about new data
  2. provide a better understanding of the data itself
- I do the first thing

Learning Methods

Three Components

1. representation language: the language used to build models
   - first-order logic
   - prior and class-conditional probability distributions
2. training algorithm: uses training examples to induce a model
   - generate clauses consistent with the examples
   - find maximum-likelihood estimates for prior and class-conditional distributions
3. prediction algorithm: uses the model to output a prediction for an observation
   - use logical deduction to make a prediction
   - use Bayes' rule to make the most probable prediction

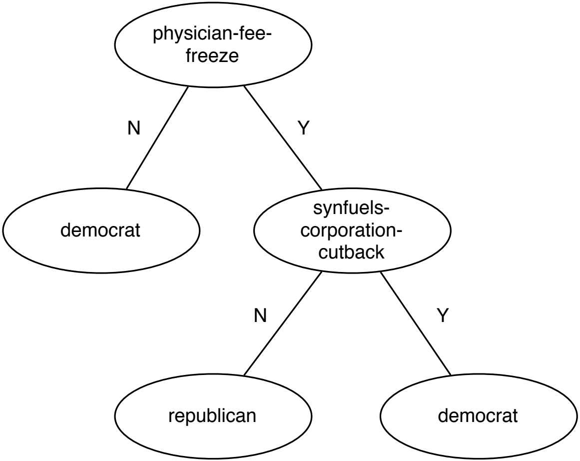
Example of Classification

- Task: predict political party based on voting record
- Data Set: 1984 US Congressional Voting Record

physician-fee-freeze	mx-missile	immigration	(12 others)	crime	party
n	n	y		n	democrat
n	n	y		y	democrat
...	...	...	...	...	...
y	y	n		y	republican
y	n	y		y	republican

- Rule: **if** (physician-fee-freeze = y) **and** (synfuels-corporation-cutback = n) **then** party = republican; **otherwise**, party = democrat

Example of a Decision Tree



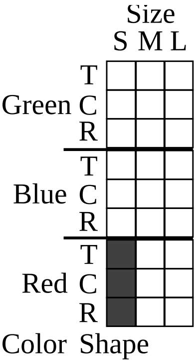
Batch versus On-line Learning

- Batch learning: When one can collect all examples for learning *before* applying the method
- Examples:
  - predict if mushrooms are poisonous (no new mushrooms)
  - predict political party based on last year's votes (all the votes have been cast)
- On-line learning: Examples arrive over time in a **stream**
- Also known as *incremental learning*
- Examples:
  - predict preferences for scheduling meetings
  - predict importance of e-mail
- What happens if the target concept changes?

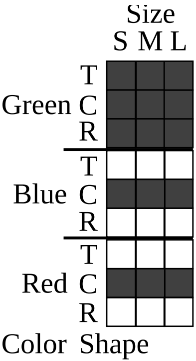
Concept Drift

- Concretely: An example has a legitimate label at one time and a different legitimate label at another time (cf. noise)
- Bayesian Decision Theory: a change in
  - the prior distribution
  - the class-conditional distribution
  - both distributions
- Geometrically: target concept in the input space changes its
  - size
  - shape
  - location
  - some combination of these
- Also known as *shifting targets*, *non-stationary environments*, *time-changing data streams*, *evolving data streams*

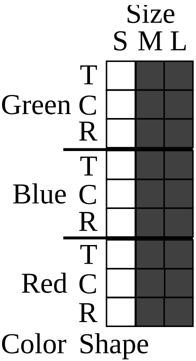
Stagger Concepts



Time steps 1–40:
`red` $\wedge$ `small`



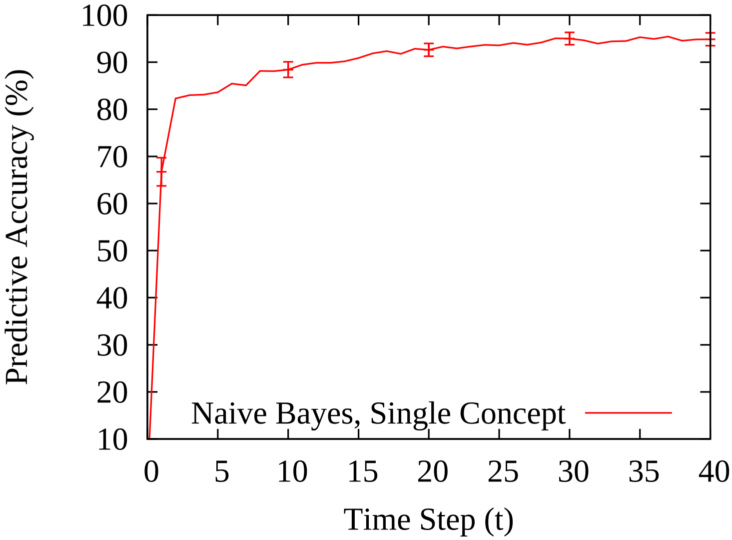
Time steps 41–80:
`green` $\vee$ `circle`



Time steps 81–120:
`medium` $\vee$ `large`

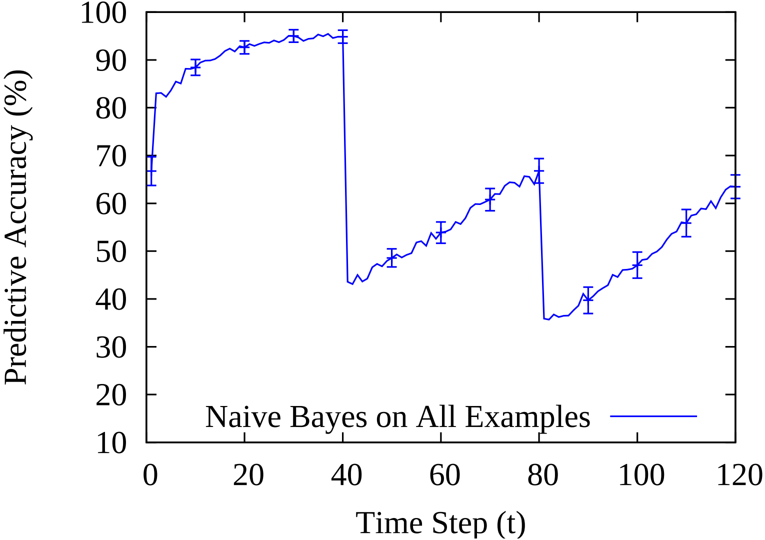
Insights into Performance

Classifier Trained on Examples from a Single Target Concept



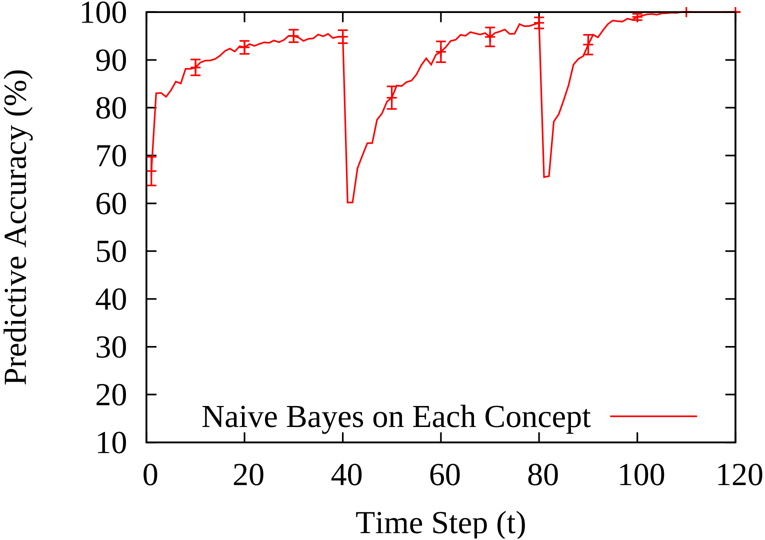
Insights into Performance

Classifier Trained on All Examples Over Three Different Target Concepts



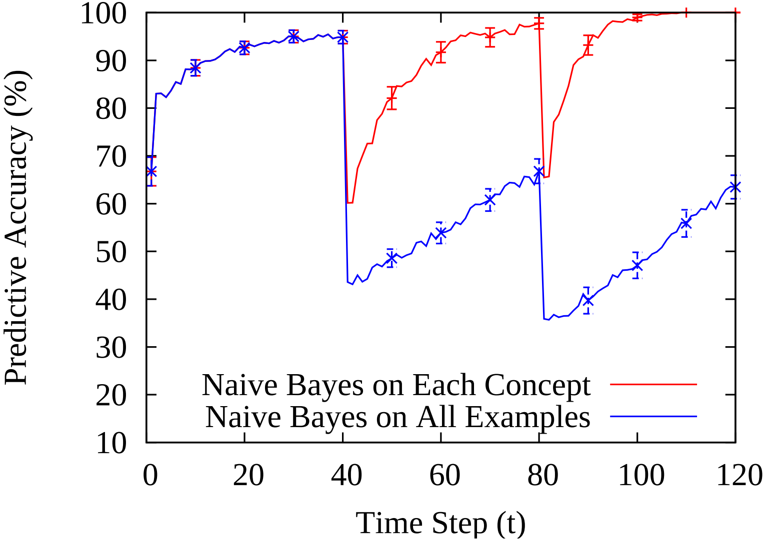
Insights into Performance

Classifier Trained on Examples from Each Target Concept

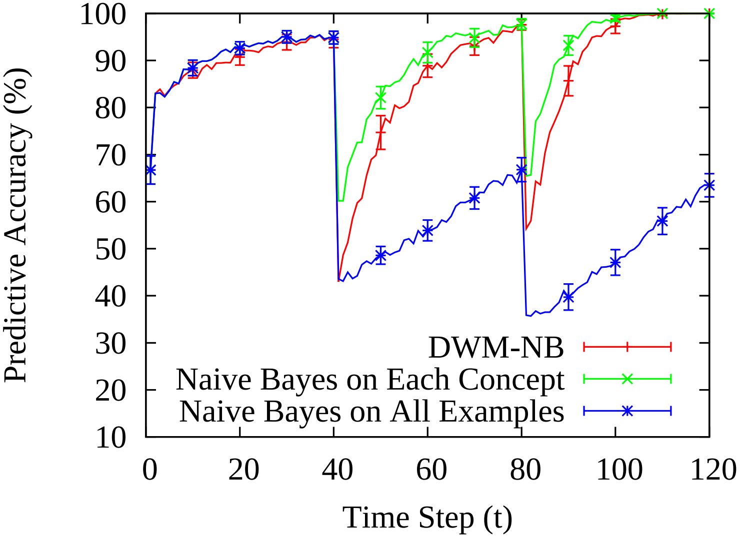


Insights into Performance

Overlay of the Previous Two Plots



Stagger Concepts

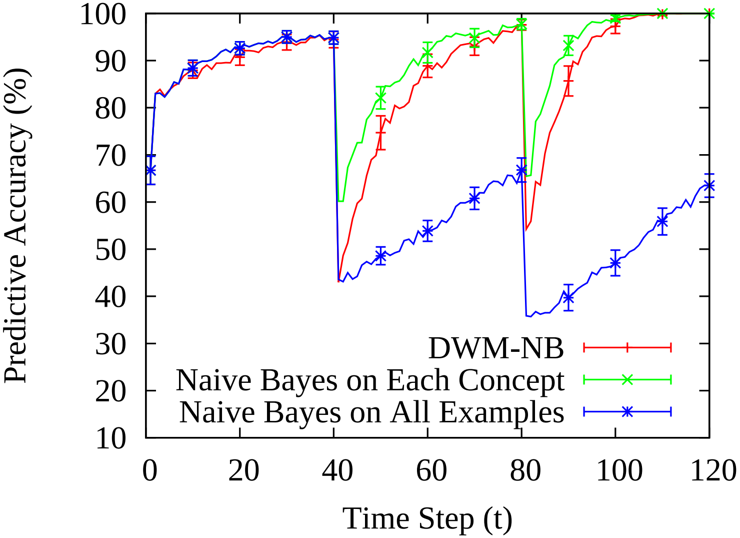


Bach and Maloof (2008)

Dynamic Weighted Majority

- Objective:
  - Build up a weighted pool of consultants
  - Make predictions based on their weighted advice
- Start with a "consultant" who is paid $1
- When presented with a decision, ask the consultant for advice
- Predict based on the advice
- If the advice is incorrect then
  - cut the consultant's pay in half
  - hire a new consultant and pay her $1
- Give feedback to all consultants so they can learn how not to be stupid
- For bad advice, cut pay and hire new consultants
- Predict based on consultants' weighted advice

Stagger Concepts



Bach and Maloof (2008)

What I Told You

- What is AI?
- Computation, Algorithms, Turing Machines, and Limits
- Logic, Probability, Numbers, and Logic and Probability
- Hypercomputation (and pseudo-hyper computation!)
- Stanley: A reason to be optimistic
- All about me!!
- Brought it on home

A Parting Shot: Tesler's Theorem

- "Intelligence is whatever machines haven't done yet."
- Commonly quoted as "AI is whatever hasn't been done yet."

Questions?

Artificial Intelligence: Everything You Need to Know in 90 Minutes

Mark Maloof

Department of Computer Science
Georgetown University
Washington, DC 20057-1232
http://www.cs.georgetown.edu/~maloof

Governing Emerging Technologies (CCTP-779)

5 October 2015

References I

S. H. Bach and M. A. Maloof. Paired learners for concept drift. In *Proceedings of the Eighth IEEE International Conference on Data Mining*, pages 23–32, Los Alamitos, CA, 2008. IEEE Press.

E. Charniak and D. McDermott. *Introduction to Artificial Intelligence*. Addison-Wesley, Reading, MA, 1985.

M. Davis. Why there is no such discipline as hypercomputation. *Applied Mathematics and Computation*, 178(1): 4–7, 2006. doi: http://dx.doi.org/10.1016%2Fj.amc.2005.09.066.

J. Haugeland. *Artificial intelligence: The very idea*. MIT Press, Cambridge, MA, 1985.

J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Reading, MA, 1979.

J. McCarthy, M. I. Minsky, N. Rochester, and C. E. Shannon. A proposal for the Dartmouth summer research project on artificial intelligence, 1955. URL `http://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html`. [Online; accessed 7 August 2014].

N. J. Nilsson. *Artificial Intelligence: A New Synthesis*. Morgan Kaufmann, San Francisco, CA, 1998.

T. Ord. Hypercomputation: Computing more than the Turing machine. Technical Report arXiv:math/0209332 [math.LO], arXiv, 2002. URL `http://arxiv.org/abs/math/0209332`. [Online; accessed 8 October 2014].

E. Rich and K. Knight. *Artificial intelligence*. McGraw-Hill, New York, NY, 2nd edition, 2009.

E. Rich, K. Knight, and S. B. Nair. *Artificial intelligence*. Tata McGraw-Hill, New Delhi, 3rd edition, 2009.

S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, NJ, 3rd edition, 2010.

J. C. Schlimmer and R. H. Granger. Beyond incremental processing: Tracking concept drift. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 502–507, Menlo Park, CA, 1986. AAAI Press.

S. Thrun. Toward robotic cars. *Communications of the ACM*, 53(4):99–106, 2010. URL `http://cacm.acm.org/magazines/2010/4/81485-toward-robotic-cars/`.

A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2(42):230–265, 1936.