Machine Learning (ML) and Knowledge Discovery in Databases (KDD)

Instructor: Rich Maclin

Course Information

- Class web page: http://www.d.umn.edu/~rmaclin/cs8751/
  - Syllabus
  - Lecture notes
  - Useful links
  - Programming assignments
- Methods for contact:
  - Email: rmaclin@d.umn.edu (best option)
  - Office: 315 HH
  - Phone: 726-8256
- Textbooks:
  - *Machine Learning*, Mitchell
- **Notes based on Mitchell's Lecture Notes**

Course Objectives

- Specific knowledge of the fields of Machine Learning and Knowledge Discovery in Databases (Data Mining)

  - Experience with a variety of algorithms
  - Experience with experimental methodology

- In-depth knowledge of several research papers

- Programming/implementation practice

- Presentation practice

Course Components

- 2 Midterms, 1 Final
  - Midterm 1 (150), February 18
  - Midterm 2 (150), April 1
  - Final (300), Thursday, May 14, 14:00-15:55 (comprehensive)
- Programming assignments (100), 3 (C++ or Java, maybe in Weka?)
- Homework assignments (100), 5
- Research Paper Implementation (100)
- Research Paper Writeup – Web Page (50)
- Research Paper Oral Presentation (50)
- Grading based on percentage (90% gets an A-, 80% B-)
  - Minimum Effort Requirement

Course Outline

- Introduction [Mitchell Chapter 1]
  - Basics/Version Spaces [M2]
  - ML Terminology and Statistics [M5]

- Concept/Classification Learning
  - Decision Trees [M3]
  - Neural Networks [M4]
  - Instance Based Learning [M8]
  - Genetic Algorithms [M9]
  - Rule Learning [M10]

Course Outline (cont)

- Unsupervised Learning

  - Clustering [Jain et al. review paper]

- Reinforcement Learning [M13]

- Learning Theory

  - Bayesian Methods [M6, Russell & Norvig Chapter]

  - PAC Learning [M7]

- Support Vector Methods [Burges tutorial]

- Hybrid Methods [M12]

- Ensembles [Opitz & Maclin paper, WF7.4]

- Mining Association Rules [Apriori paper]

What is Learning?

Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the same task or tasks drawn from the same population more effectively the next time. -- Simon, 1983

Learning is making useful changes in our minds. -- Minsky, 1985

Learning is constructing or modifying representations of what is being experienced. -- McCarthy, 1968

Learning is improving automatically with experience. -- Mitchell, 1997

Why Machine Learning?

- ## Data, Data, DATA!!!
  - Examples
    - World wide web
    - Human genome project
    - Business data (WalMart sales "baskets")
  - Idea: sift heap of data for nuggets of knowledge
- ## Some tasks beyond programming
  - Example: driving
  - Idea: learn by doing/watching/practicing (like humans)
- ## Customizing software
  - Example: web browsing for news information
  - Idea: observe user tendencies and incorporate

Typical Data Analysis Task

Data:

PatientId=103: **EmergencyC-Section=yes**

Age=23, Time=53, FirstPreg?=no, Anemia=no, Diabetes=no, PrevPremBirth=no, UltraSound=?, ElectiveC-Section=?

Age=23, Time=105, FirstPreg?=no, Anemia=no, Diabetes=yes, PrevPremBirth=no, UltraSound=abnormal, ElectiveC-Section=no

Age=23, Time=125, FirstPreg?=no, Anemia=no, Diabetes=yes, PrevPremBirth=no, UltraSound=?, ElectiveC-Section=no

PatientId=231: **EmergencyC-Section=no**

Age=31, Time=30, FirstPreg?=yes, Anemia=no, Diabetes=no, PrevPremBirth=no, UltraSound=?, ElectiveC-Section=?

Age=31, Time=91, FirstPreg?=yes, Anemia=no, Diabetes=no, PrevPremBirth=no, UltraSound=normal, ElectiveC-Section=no

…

Given

- 9714 patient records, each describing a pregnancy and a birth
- Each patient record contains 215 features (some are unknown)

Learn to predict:

- Characteristics of patients at high risk for Emergency C-Section

Credit Risk Analysis

Data:

> **ProfitableCustomer=No,** CustId=103, YearsCredit=9, LoanBalance=2400,
> Income=52,000, OwnHouse=Yes, OtherDelinqAccts=2,
> MaxBillingCyclesLate=3
>
> **ProfitableCustomer=Yes,** CustId=231, YearsCredit=3, LoanBalance=500,
> Income=36,000, OwnHouse=No, OtherDelinqAccts=0,
> MaxBillingCyclesLate=1
>
> **ProfitableCustomer=Yes,** CustId=42, YearsCredit=15, LoanBalance=0,
> Income=90,000, OwnHouse=Yes, OtherDelinqAccts=0,
> MaxBillingCyclesLate=0
>
> …

Rules that might be learned from data:

```
IF Other-Delinquent-Accounts > 2, AND
   Number-Delinquent-Billing-Cycles > 1
THEN Profitable-Customer? = No   [Deny Credit Application]
IF Other-Delinquent-Accounts == 0, AND
   ((Income > $30K) OR (Years-of-Credit > 3))
THEN Profitable-Customer? = Yes  [Accept Application]
```

Analysis/Prediction Problems

- What kind of direct mail customers buy?

- What products will/won't customers buy?

- What changes will cause a customer to leave a bank?

- What are the characteristics of a gene?

- Does a picture contain an object (does a picture of space contain a metereorite -- especially one heading towards us)?

- … Lots more

Tasks too Hard to Program
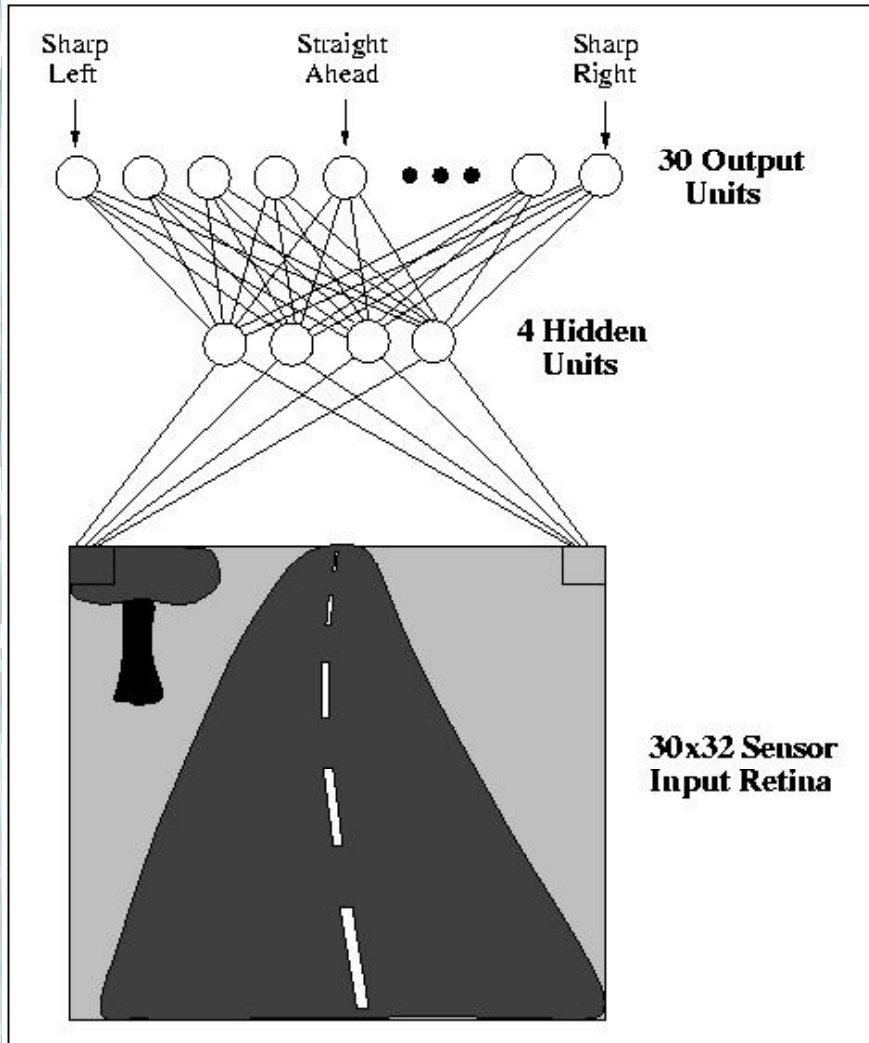
ALVINN [Pomerleau] drives
70 MPH on highways

Software that Customizes to User

Defining a Learning Problem

Learning = improving with experience at some task

- improve over task *T*
- with respect to performance measure *P*
- based on experience *E*

Ex 1: Learn to play checkers

T: play checkers

P: % of games won

E: opportunity to play self

Ex 2: Sell more CDs

T: sell CDs

P: # of CDs sold

E: different locations/prices of CD

Key Questions

T: play checkers, sell CDs

P: % games won, # CDs sold

To generate machine learner need to know:

- What experience?
  - Direct or indirect?
  - Learner controlled?
  - Is the experience representative?
- What exactly should be learned?
- How to represent the learning function?
- What algorithm used to learn the learning function?

Types of Training Experience

Direct or indirect?

Direct - observable, measurable

- sometimes difficult to obtain

  - Checkers - is a move the best move for a situation?

- sometimes straightforward

  - Sell CDs - how many CDs sold on a day? (look at receipts)

Indirect - must be inferred from what is measurable

- Checkers - value moves based on outcome of game

- *Credit assignment problem*

Types of Training Experience (cont)

Who controls?

- – Learner - what is best move at each point? (Exploitation/Exploration)
- – Teacher - is teacher's move the best? (Do we want to just emulate the teachers moves??)

BIG Question: is experience *representative* of performance goal?

- – If Checkers learner only plays itself will it be able to play humans?
- – What if results from CD seller influenced by factors not measured (holiday shopping, weather, etc.)?

Choosing Target Function

Checkers - what does learner do - make moves

    ChooseMove - select move based on board

        $ChooseMove : Board \rightarrow Move$

        $V : Board \rightarrow \Re$

    ChooseMove(b): from *b* pick move with highest value

    But how do we define *V(b)* for boards b?

    Possible definition:

        V(b) = 100 if *b* is a final board state of a win

        V(b) = -100 if *b* is a final board state of a loss

        V(b) = 0 if *b* is a final board state of a draw

        if *b* not final state, *V(b)* = *V(b´)* where *b´* is best final board reached by starting at *b* and playing optimally from there

        Correct, but not operational

Representation of Target Function

- Collection of rules?

  IF double jump available THEN

  make double jump

- Neural network?

- Polynomial function of problem features?

$$w_0 + w_1 \# blackPieces(b) + w_2 \# redPieces(b) +$$
$$w_3 \# blackKings(b) + w_4 \# redKings(b) +$$
$$w_5 \# redThreatened(b) + w_6 \# blackThreatened(b)$$

Obtaining Training Examples

$V(b)$: the true target function

$\hat{V}(b)$: the learned function

$V_{train}(b)$: the training value

One rule for estimating training values :

$$V_{train}(b) \leftarrow \hat{V}(Successor(b))$$

Choose Weight Tuning Rule

LMS Weight update rule:

Do repeatedly :

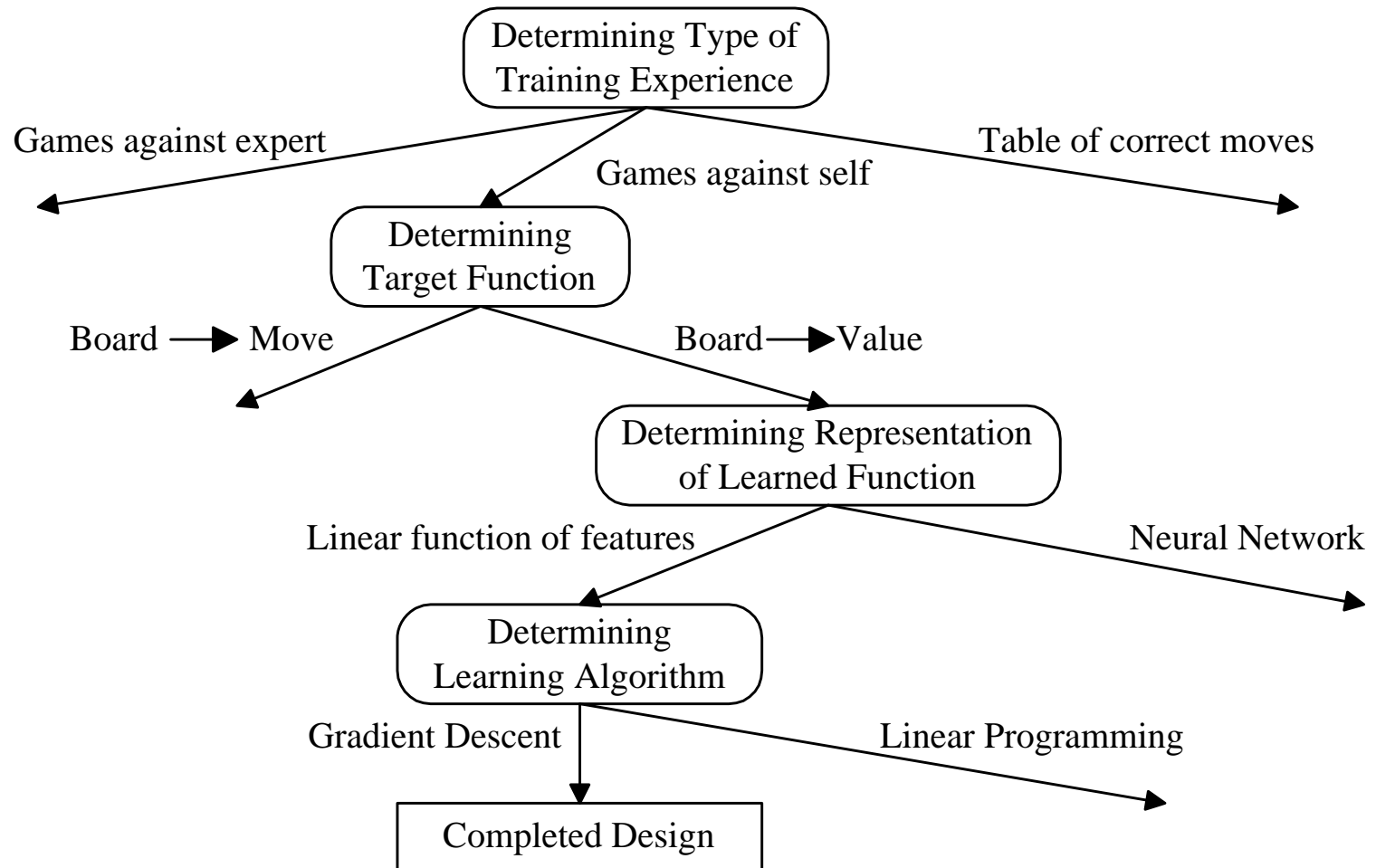Select a training example b at random

1. Compute $error(b)$:

$$error(b) = V_{train}(b) - \hat{V}(b)$$

2. For each board feature f_i, update weight w_i :

$$w_i \leftarrow w_i + c \times f_i \times error(b)$$

c is some small constant, say 0.1, to moderate

rate of learning

Design Choices

Determining Type of
Training Experience

Games against expert

Games against self

Table of correct moves

Determining
Target Function

Board → Move

Board → Value

Determining Representation
of Learned Function

Linear function of features

Neural Network

Determining
Learning Algorithm

Gradient Descent

Linear Programming

Completed Design

Some Areas of Machine Learning

- **Inductive Learning**: inferring new knowledge from observations (not guaranteed correct)
  - Concept/Classification Learning - identify characteristics of class members (e.g., what makes a CS class fun, what makes a customer buy, etc.)
  - Unsupervised Learning - examine data to infer new characteristics (e.g., break chemicals into similar groups, infer new mathematical rule, etc.)
  - Reinforcement Learning - learn appropriate moves to achieve delayed goal (e.g., win a game of Checkers, perform a robot task, etc.)
- **Deductive Learning**: recombine existing knowledge to more effectively solve problems

Classification/Concept Learning
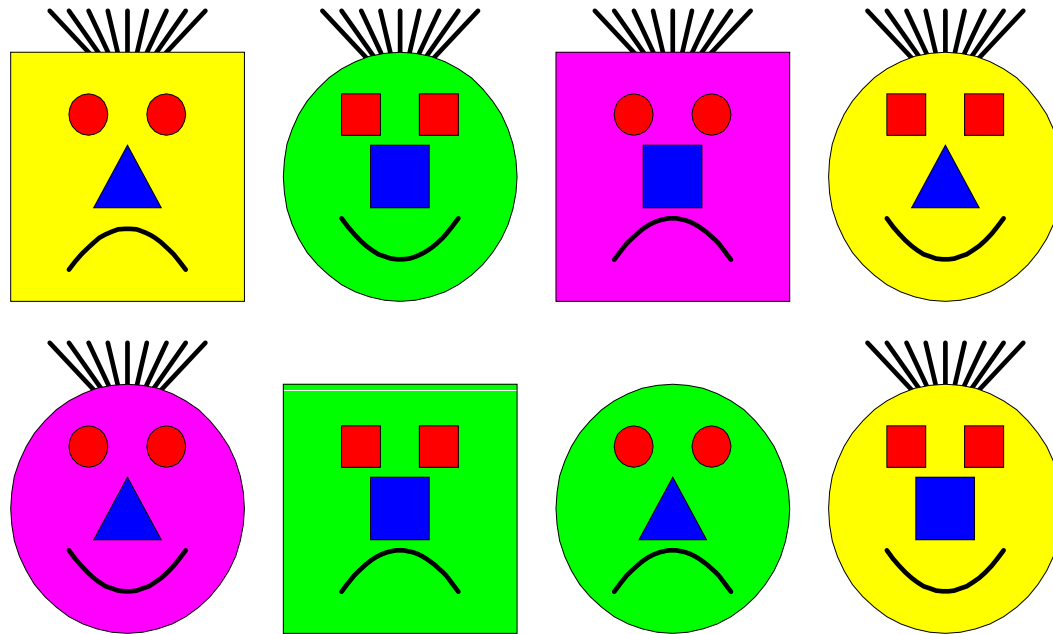
- What characteristic(s) predict a smile?
  - Variation on Sesame Street game: *why are these things a lot like the others (or not)?*

- ML Approach: infer model (characteristics that indicate) of why a face is/is not smiling
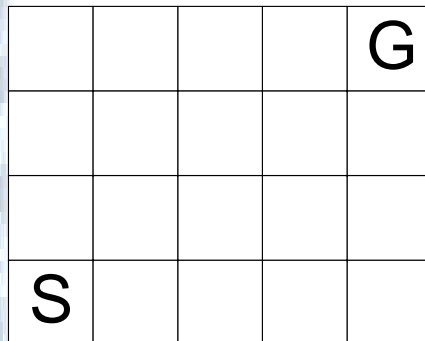
Unsupervised Learning
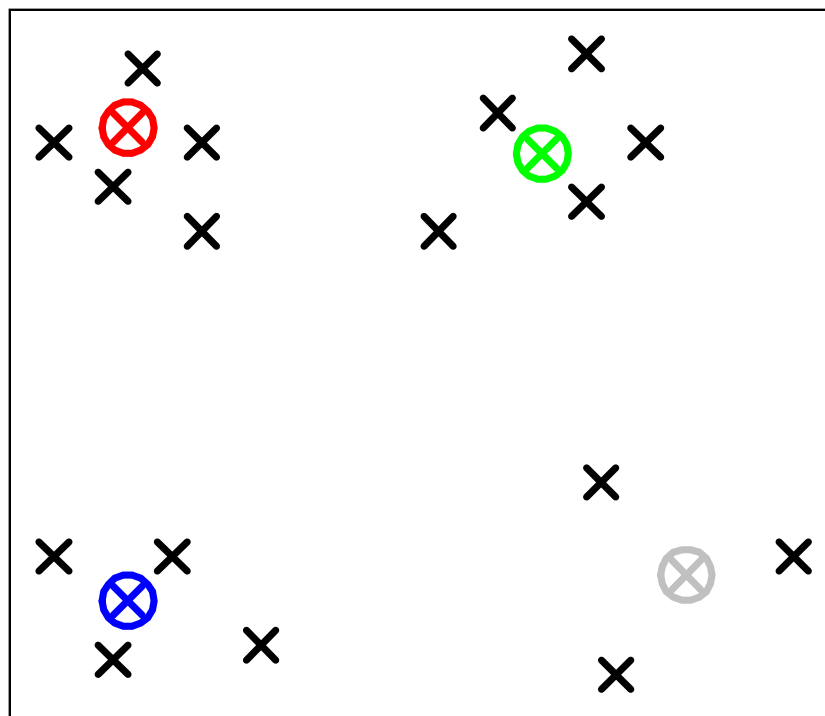
- Clustering - group points into "classes"
- Other ideas:
  - look for mathematical relationships between features
  - look for anomalies in data bases (data that does not fit)

Reinforcement Learning

Problem   **Policy**

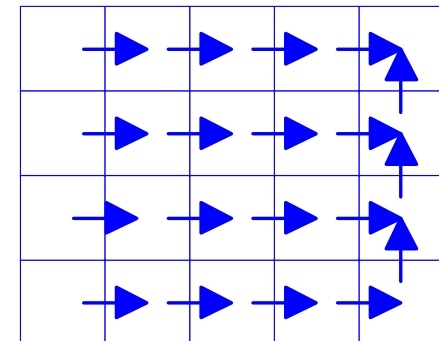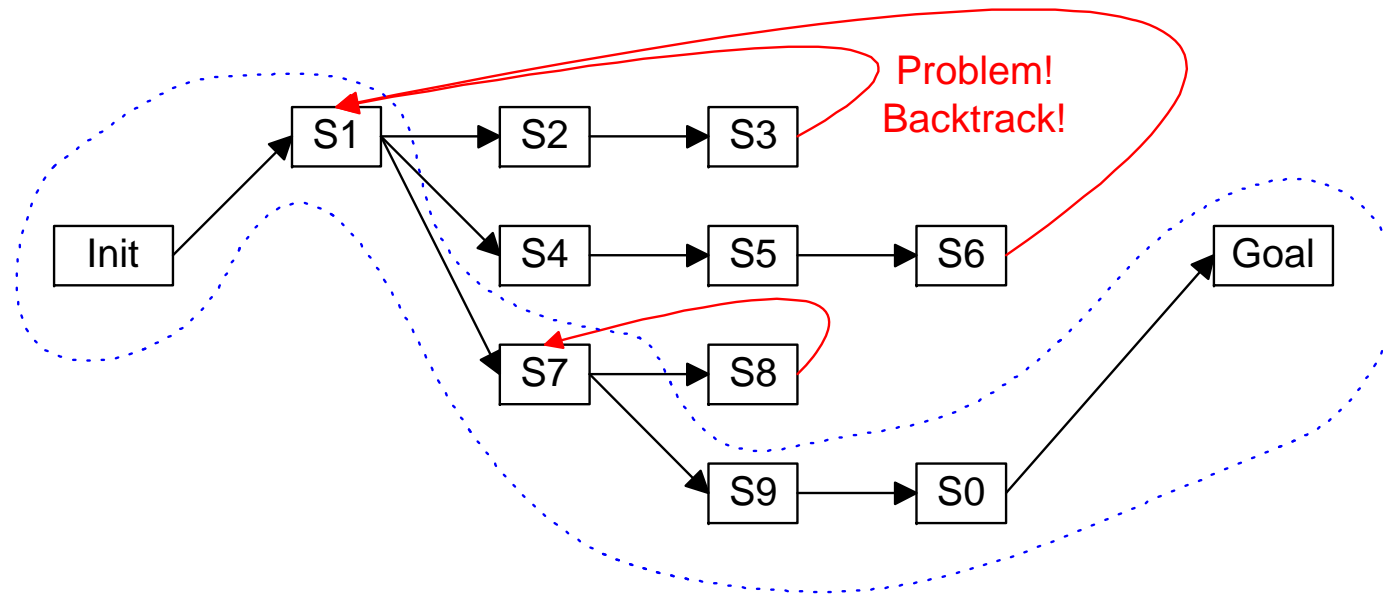|			
			G
S			

S - start
G - goal
Possible actions:
↑ up   left ←
↓ down   right →

- Problem: feedback (reinforcements) are delayed - how to value intermediate (no goal states)

- Idea: online dynamic programming to produce policy function

- Policy: action taken leads to highest future reinforcement (if policy followed)

Analytical Learning

- During search processes (planning, etc.) remember work involved in solving tough problems

- Reuse the acquired knowledge when presented with similar problems in the future (avoid bad decisions)

The Present in Machine Learning

The tip of the iceberg:

- First-generation algorithms: neural nets, decision trees, regression, **support vector machines**, **kernel methods**, **Bayesian networks**,…

- Composite algorithms - ensembles

- Significant work on assessing effectiveness, limits

- Applied to simple data bases

- Budding industry (especially in data mining)

The Future of Machine Learning

Lots of areas of impact:

- Learn across multiple data bases, as well as web and news feeds

- Learn across multi-media data

- Cumulative, lifelong learning

- Agents with learning embedded

- Programming languages with learning embedded?

- Learning by active experimentation

What is Knowledge Discovery in Databases (i.e., Data Mining)?

- Depends on who you ask

- General idea: the analysis of large amounts of data (and therefore efficiency is an issue)

- Interfaces several areas, notably machine learning and database systems

- Lots of perspectives:
  - ML: learning where efficiency matters
  - DBMS: extended techniques for analysis of raw data, automatic production of knowledge

- What is all the hubbub?
  - Companies make lots of money with it (e.g., WalMart)

Related Disciplines

- Artificial Intelligence
- Statistics
- Psychology and neurobiology
- Bioinformatics and Medical Informatics
- Philosophy
- Computational complexity theory
- Control theory
- Information theory
- Database Systems
- ...

Issues in Machine Learning

- What algorithms can approximate functions well (and when)?
- How does number of training examples influence accuracy?
- How does complexity of hypothesis representation impact it?
- How does noisy data influence accuracy?
- What are the theoretical limits of learnability?
- How can prior knowledge of learner help?
- What clues can we get from biological learning systems?